

---

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

Arcabouço de Software para a Aquisição de Dados  
de Saúde Através de Jogos Eletrônicos

Antônio Dias dos Santos Júnior

Dissertação submetida à Coordenação do Curso de Pós-Graduação em  
Ciência da Computação da Universidade Federal de Campina Grande -  
Campus I como parte dos requisitos necessários para obtenção do grau  
de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Engenharia de Software

Angelo Perkusich (Orientador)

Hygo Oliveira de Almeida (Orientador)

Campina Grande, Paraíba, Brasil

©Antônio Dias dos Santos Júnior, Fevereiro de 2013

---

DIGITALIZAÇÃO:  
SISTEMOTECA - UFCG

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

S237a Santos Júnior, Antonio Dias dos.  
Arcabouço de software para a aquisição de dados de saúde através de jogos eletrônicos / Antonio Dias dos Santos Júnior. -- 2013.  
63 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

"Orientação: Prof. Dr. Angelo Perkusich, Prof. Dr. Hyggo Oliveira de Almeida".  
Referências.

1. Jogos Eletrônicos. 2. Monitoramento. 3. Assistência Médica Pervasiva. I. Perkusich, Angelo. II. Almeida, Hyggo Oliveira de. III. Título.

CDU 004.4(043)

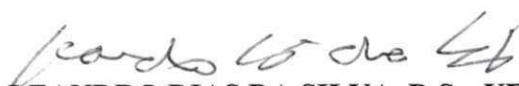
**"ARCABOUÇO DE SOFTWARE PARA A AQUISIÇÃO DE DADOS DE SAÚDE ATRAVÉS  
DE JOGOS ELETRÔNICOS"**

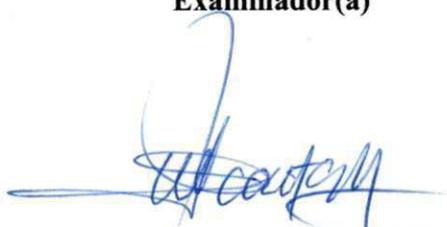
**ANTONIO DIAS DOS SANTOS JUNIOR**

**DISSERTAÇÃO APROVADA EM 15/03/2013**

  
**HYGGO OLIVEIRA DE ALMEIDA, D.Sc, UFCG**  
**Orientador(a)**

  
**ANGELO PERKUSICH, D.Sc, UFCG**  
**Orientador(a)**

  
**LEANDRO DIAS DA SILVA, D.Sc, UFAL**  
**Examinador(a)**

  
**MARCOS RICARDO ALCÂNTARA MORAIS, D.Sc, UFCG**  
**Examinador(a)**

**CAMPINA GRANDE - PB**

## Resumo

Os avanços alcançados na área de tecnologia da informação e comunicação trouxeram a possibilidade de realizar em casa o monitoramento da saúde, o que antes só podia ser feito em locais especializados, como clínicas e hospitais. O monitoramento realizado no conforto da casa dos indivíduos permite prover serviços para um maior número de pessoas, otimizando o uso de recursos humanos e financeiros. Porém, este sistema de monitoramento enfrenta problemas, sendo o principal deles a dificuldade de uso e a falta de motivação dos usuários. Embora já exista a possibilidade e viabilidade em usar sensores para monitoramento, essa é uma atividade que ainda não está integrada ao cotidiano das pessoas.

Para resolver o problema de falta de motivação nos usuários, jogos eletrônicos vêm sendo fortemente utilizados como ferramenta motivadora. Entretanto, os jogos que tentam motivar o jogador a exercitar-se ou a ser monitorado geralmente requerem que o jogador habitue-se com um estilo de jogo que não é parte de seu cotidiano. As ações realizadas dentro do jogo não são parte de seus jogos preferidos.

Neste trabalho apresenta-se um arcabouço de software para monitorar dados de saúde através de jogos eletrônicos. Sua forma de captura de dados permite criar jogos para monitoramento ainda mantendo a jogabilidade. A validação do arcabouço se deu com o desenvolvimento de dois jogos, os quais foram testados com 18 pessoas na faixa de idade de 19 a 27 anos do Laboratório Embedded, da Universidade Federal de Campina Grande, e do Instituto Federal de Alagoas. Os resultados indicaram que os jogos foram considerados divertidos pela maioria das pessoas, que também declararam que os incluíam em sua rotina diária. Além disso, foi possível coletar os dados de saúde de maneira a utilizá-los para a detecção de doenças motoras.

## **Abstract**

Health monitoring, which could only be done in specialized places, such as clinics and hospitals, can now be performed at home thanks to the advances in Information and Communication Technologies. Monitoring health data in the comfort of one's homes allows the provision of services to a greater number of people, optimizing human and financial resources. However, this system faces problems such as difficulty of use and lack of motivation in users. Although it is feasible to use sensors to monitor people, this is still something not easy to integrate to their daily lives.

To tackle the lack of motivation among people who need to be monitored, electronic games have been extensively used as a motivational tool. However, games that intend to motivate the player to exercise or to be monitored usually have a different gameplay than what the player would enjoy.

This work presents a software framework to monitor health data through electronic games, allowing it to be acquired while keeping the playability. The framework was used for the development of two games, tested by 18 people with ages ranging from 19 to 27, from the Embedded Laboratory at the Federal University of Campina Grande, and Federal Institute of Alagoas. Results showed that most people found the games to be funny, and also stated that they would include the games in their daily routine. Also, it was possible to collect health data to be used in the detection of motor diseases.

## **Agradecimentos**

Agradeço aos meus pais e família, pois devo a eles tudo o que alcancei e me tornei. À minha namorada, Mayelli, pela paciência, cumplicidade e apoio incondicionais durante todo o tempo em que estivemos juntos. Obrigado por me entender nos momentos em que mais necessito e por estar sempre ao meu lado. Aos amigos que fiz em Campina Grande, estudantes como eu, que dividiram comigo os momentos felizes e não tão felizes da vida acadêmica. Agradeço especialmente a Leonardo e Rafael, por sua contribuição direta, indispensável para a realização deste trabalho. Também a Romeryto e Maurílio, grandes amigos e companheiros de sala. Às funcionárias da COPIN, por sua disposição a sempre ajudar nós alunos da pós-graduação. Aos meus orientadores Angelo e Hyggo, por sua participação essencial neste trabalho, especialmente pela paciência, auxílio e idéias valiosas. À CAPES, pelo apoio financeiro.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Problemática . . . . .	3
1.2	Objetivo . . . . .	6
1.3	Relevância . . . . .	7
1.4	Organização . . . . .	8
<b>2</b>	<b>Fundamentação Teórica</b>	<b>9</b>
2.1	Assistência Médica Pervasiva ( <i>Pervasive Healthcare</i> ) . . . . .	9
2.2	Jogos Sérios . . . . .	10
2.3	<i>Exergames</i> . . . . .	11
2.4	Sensores para Captura de Dados de Saúde . . . . .	12
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>14</b>
3.1	Video Games in Health Care: Closing the Gap . . . . .	14
3.2	Adaptive Virtual Reality Games for Rehabilitation of Motor Disorders . . . . .	15
3.3	Context Aware Serious Games Framework for Sport and Health . . . . .	16
3.4	Serious Game Based on First Aid Education for Individuals with Autism Spectrum Disorder (ASD) Using Android Mobile Services . . . . .	19
3.5	Game based Learning to Enhance Capabilities of Elderly People . . . . .	20
3.6	Conclusão . . . . .	20
<b>4</b>	<b>Arcabouço para a Aquisição de Dados de Saúde Utilizando Jogos Eletrônicos</b>	<b>21</b>
4.1	Requisitos do Arcabouço . . . . .	21
4.2	Arquitetura . . . . .	22
4.2.1	Módulo Gerenciador de Dados . . . . .	25

---

4.2.2	Analisador de Dados . . . . .	30
4.3	Utilizando e Estendendo o Arcabouço . . . . .	34
4.3.1	Lendo e Armazenando os Dados de um Sensor Durante um Jogo . . . . .	34
4.3.2	Estendendo o Arcabouço . . . . .	36
4.3.3	<i>Classes de Análise de Dados</i> . . . . .	37
4.3.4	<i>Filtros</i> . . . . .	38
4.3.5	<i>Gerenciador de Arquivos</i> . . . . .	38
4.3.6	<i>Classes de Escrita de Dados</i> . . . . .	39
4.4	Conclusão . . . . .	40
<b>5</b>	<b>Estudo de Caso</b>	<b>41</b>
5.1	<i>Pinball World</i> . . . . .	41
5.1.1	Desenvolvimento do Jogo . . . . .	42
5.2	<i>Catch the Spheres</i> . . . . .	44
5.2.1	Desenvolvimento do Jogo . . . . .	47
5.3	Avaliação Experimental . . . . .	49
5.4	Conclusão . . . . .	55
<b>6</b>	<b>Considerações Finais</b>	<b>56</b>

# Lista de Figuras

1.1	Kinect, WiiMote e PsMove, de cima para baixo . . . . .	4
2.1	Os jogos Wii Fit e Wii Sports . . . . .	12
3.1	Arquitetura do sistema de realidade virtual para reabilitação física [24] . . .	16
3.2	Dispositivos de entrada . . . . .	17
3.3	<i>SocialAware</i> . . . . .	18
4.1	Transferência de Dados Dentro do Arcabouço . . . . .	23
4.2	Arquitetura do Arcabouço . . . . .	23
4.3	Diagrama de Classes do Arcabouço . . . . .	25
4.4	Eixos do acelerômetro . . . . .	32
4.5	O <i>Skeleton Space</i> . . . . .	33
4.6	Plotagem das coordenadas da mão esquerda . . . . .	33
5.1	O Jogo <i>Pinball World</i> . . . . .	42
5.2	Plotagem das coordenadas da mão esquerda . . . . .	46
5.3	O jogo <i>Catch the Spheres</i> . . . . .	46
5.4	Respostas do questionário . . . . .	54

## Lista de Tabelas

4.1	Operações disponibilizadas pelo <i>web service</i> . . . . .	24
5.1	Tipos de tremor, diferenciados pela frequência, amplitude e início em relação a movimentos voluntários . . . . .	45
5.2	O Questionário GQM . . . . .	50
5.3	Variação de Velocidade em Duas Sessões do Jogo <i>Catch the Spheres</i> . . . . .	53

## Lista de Códigos Fonte

4.1	Um exemplo de formato do arquivo de regras . . . . .	33
4.2	O objeto JSON <i>leitura</i> . . . . .	35
4.3	Coleta e envio das leituras . . . . .	36
4.4	Um Exemplo de Classe de Análise de Dados . . . . .	37
4.5	Um Exemplo de Classe de Filtragem de Dados . . . . .	38
4.6	Um Exemplo de Extensão do Gerenciador de Arquivos . . . . .	38
4.7	Um Exemplo de Classe de Escrita de Dados . . . . .	39
5.1	O Método <code>Update</code> . . . . .	43
5.2	O Método <code>enviarLeituras</code> . . . . .	43
5.3	O Método <code>UpdatePosition</code> . . . . .	47
5.4	O Método <code>mudarDeCor</code> . . . . .	48
5.5	Registro de eventos do jogo . . . . .	48
5.6	Registro de posições do jogador durante jogo . . . . .	49

# Capítulo 1

## Introdução

Os gastos com saúde totalizam 9% do PIB no Brasil e 16,2% nos Estados Unidos<sup>1</sup>. Os custos crescentes com assistência à saúde devido ao aumento na população com mais de 60 anos nos países mostram a importância de gerenciar recursos, focando no conforto do indivíduo [8]. Graças aos avanços alcançados na área de tecnologia da informação e comunicação, conceitos promissores como assistência médica pervasiva (*Pervasive Healthcare*) surgiram com potencial para auxiliar o atual sistema de assistência à saúde [44]. A assistência médica pervasiva é um avanço na assistência à saúde para qualquer pessoa, a qualquer hora e lugar, assistida por dispositivos computacionais integrados entre si e ao cotidiano das pessoas [10]. Um desafio importante é prover essa qualidade de serviços para um número crescente de pessoas, otimizando o uso de recursos humanos e financeiros. O grande diferencial no uso de dispositivos inteligentes, interfaces com o usuário, sensores corporais e dispositivos de comunicação sem fio dentro do contexto de assistência médica consiste no monitoramento constante do estado de saúde das pessoas, mesmo fora do hospital. Apesar desse monitoramento constante não ser capaz de eliminar completamente os erros médicos, existe um grande potencial de mitigar alguns dos problemas relacionados com a falta de informação [43].

O monitoramento baseado em sensores que capturam dados de saúde dos indivíduos pode ser importante, por exemplo, para detecção prematura de doenças, ajudando tanto na prevenção destas quanto na administração de um melhor tratamento, podendo prover maior qualidade de vida para um indivíduo. Formas de avaliação da saúde que podem ser reali-

---

<sup>1</sup><https://www.cia.gov/library/publications/the-world-factbook/>

---

zadas através do monitoramento são sugeridas em [27], como a determinação do intervalo normal de cada dado capturado para cada indivíduo, a análise estatística da correlação entre os dados de saúde e a predição de tendências relativas à saúde no futuro. Por fim, a análise dos dados capturados pode ajudar a identificar manifestações de sintomas que relacionam-se com a ocorrência de um problema de saúde, que podem ser vistos como um sinal primário da doença, se ocorrerem repetidamente. A acumulação de tais resultados dentro de uma população pode ser usada não somente para diagnosticar mais cedo, mas também para identificar a origem da doença em estudos epidemiológicos.

Apesar de todas as vantagens mencionadas e utilidades atribuídas aos sistemas de monitoramento, existem alguns problemas a serem mencionados. Dentre eles estão a dificuldade de uso e a falta de motivação dos usuários. Atualmente, dados de saúde capturados durante o monitoramento, tais como pressão sanguínea, frequência cardíaca e temperatura, por exemplo, são sincronizados com um servidor externo para que sejam armazenados e/ou analisados por profissionais de saúde [19; 26; 28]. Os sistemas pessoais de monitoramento de saúde requerem que o usuário tenha disciplina durante a leitura dos seus dados de saúde, ou que ao menos esteja utilizando os sensores corretamente para obter uma leitura satisfatória. Desta forma, os mecanismos de monitoramento tradicionais podem ser considerados entediantes pelas pessoas [22]. Certos métodos podem ser repetitivos e cansativos, exigindo que sensores sejam acoplados ao corpo, tornando complicada a obtenção da colaboração ativa do indivíduo durante seu monitoramento. Há também a resistência de indivíduos ao monitoramento realizado em casa [1], uma vez que certos tipos de monitoramento realizados no lar requerem uma adaptação do ambiente. É necessária uma forma que estimule os indivíduos a incluir o monitoramento na sua rotina, tornando-os participantes mais ativos do seu próprio tratamento, tornando este mais motivador e efetivo.

Embora já seja possível e viável a utilização de dispositivos para monitorar os usuários, essa tarefa ainda não está integrada ao seu cotidiano. Nesse contexto, jogos eletrônicos vêm sendo fortemente utilizados como ferramenta motivadora. Com as novas possibilidades de interação trazidas pelas novas tecnologias usadas em jogos para captura de movimento, é possível inserir elementos inovadores no projeto de jogos eletrônicos. Dispositivos e sensores, como acelerômetros, giroscópios, câmeras, tapetes, esteiras e sensores de pressão, são utilizados no contexto de jogos para criar novas formas de interação com o usuário [37].

É neste contexto de utilização de jogos para viabilizar um monitoramento de saúde de usuários de forma motivadora e efetiva que se insere este trabalho. A abordagem aqui proposta visa permitir a coleta de dados de saúde de indivíduos saudáveis utilizando jogos, de forma a antecipar cenários de doença ou de complicações de saúde através da identificação de anormalidades entre os dados coletados.

## 1.1 Problemática

A ideia de usar elementos de projeto de jogos, como recompensas, desafios e elementos gráficos, dentro de contextos diferentes para motivar e aumentar a atividade e retenção de usuários rapidamente ganhou ênfase. Isso tem gerado intenso debate e também está presente em inúmeras aplicações, como finanças, saúde e educação [15].

As aplicações na saúde que usam elementos de jogos são dirigidas a aumentar a motivação de indivíduos em três áreas. Primeiro, para aumentar o empenho de um indivíduo em aprender mais sobre sua condição e tratamento. Segundo, para ser usado como uma ferramenta terapêutica de distração contra a dor e ansiedade. E, por fim, para encorajar indivíduos jovens a continuar com longos tratamentos [45].

As primeiras aplicações de jogos no contexto de saúde ocorreram com o desenvolvimento de uma interface inovadora, de forma que jogos que tipicamente não exigiam nenhum esforço físico passaram a ser usados para motivar adultos e adolescentes a engajar-se em terapias e atividades físicas [22].

Os avanços no desenvolvimento de interfaces alcançados pela indústria de jogos incluem tecnologias que fazem um rastreamento espacial, permitindo ao jogador controlar o ambiente do jogo com movimentos do seu corpo. As plataformas mais conhecidas atualmente são o Kinect, para o Xbox 360 da Microsoft; o Wii, da Nintendo; o EyeToy, usado no PlayStation2 da Sony, e o Move, para PlayStation3. As plataformas EyeToy e Kinect utilizam câmeras para capturar o movimento do jogador, enquanto que o Move e Wii utilizam uma combinação de acelerômetros, giroscópios e sensores infravermelho [33]. Na Figura 1.1, ilustra-se o Kinect, o controle do Wii e o PsMove.

Essas novas formas de interação com jogos digitais requerem naturalmente que o jogador seja mais ativo fisicamente durante o curso do jogo. Com essa inovação surgiram os

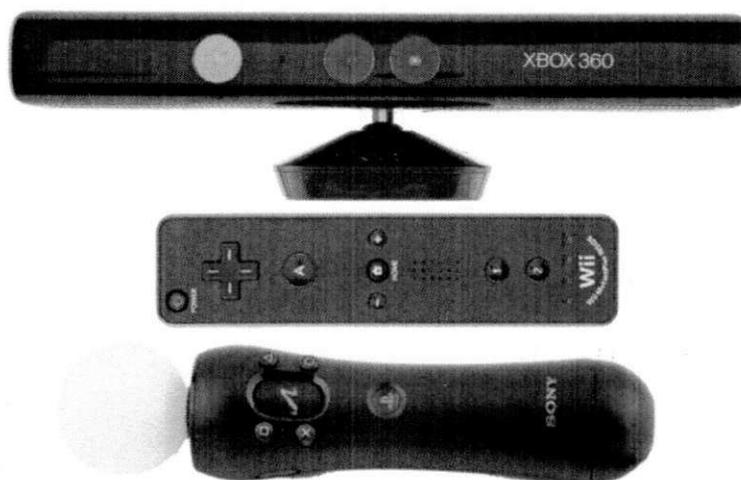


Figura 1.1: Kinect, WiiMote e PsMove, de cima para baixo

*Exergames* [37], cujo nome é um termo criado a partir da combinação de *exercise* e *digital gaming*, estimulando o jogador a empenhar-se no desenvolvimento de habilidades motoras durante o jogo. *Exergames* foram primariamente projetados para promover o gasto calórico e elevar os batimentos cardíacos através de atividade aeróbica, isto é, motivando a prática de exercícios durante o curso de um jogo eletrônico, uma vez que nem todos os indivíduos têm a motivação para praticar exercícios necessários para evitar a ocorrência de doenças ou diminuir sua progressão. Dispositivos e sensores como acelerômetros, giroscópios, câmeras, tapetes, esteiras e sensores de pressão são utilizados no contexto de exergames para medir a interação com o usuário. Alguns dos exergames comerciais mais conhecidos são o Dance Dance Revolution e o Wii Fit. Apesar de esses jogos não serem destinados primariamente à melhoria da saúde, foram constatados efeitos positivos do seu uso [7; 21].

Alguns jogos que se inserem no contexto de saúde são classificados como jogos de diagnóstico médico, que são jogos usados para examinar um indivíduo sob a suspeita de que já possuem algum problema de saúde. Para jogos de diagnóstico funcionarem corretamente, técnicas e pesquisas adequadas devem ser aplicadas ao mecanismo central do jogo para diagnosticar o indivíduo com precisão. Uma das questões mais importantes a se tratar em jogos para diagnóstico é manter a coleta de dados discreta e o nível de estresse do indivíduo baixo. Um indivíduo que está ansioso e sob pressão por estar sendo submetido a vários exames

pode não prover resultados precisos e, portanto, os tratamentos seguintes podem não ser tão efetivos. Por isso, jogos para diagnóstico devem ser mantidos em um nível casual, para que façam o indivíduo sentir-se como se não estivesse sendo examinado [9].

Para que aplicações direcionadas ao monitoramento da saúde sejam possíveis, a aquisição de dados dos jogadores pode ser feita com o uso de sensores e dispositivos de entrada diversos, como, por exemplo, um controle que capte movimentos através de acelerômetros. As novas possibilidades de interação desenvolvidas pela indústria de jogos vão além dos métodos convencionais de entrada, podendo ser usadas para avaliar as capacidades motoras e cognitivas de um jogador [7].

O problema é que as abordagens já propostas de jogos para a saúde têm como objetivo principal a execução de exercícios específicos. Neste caso, o indivíduo ainda se sente em tratamento ou sob monitoramento, podendo apresentar a mesma resistência que existe no caso do monitoramento tradicional. Os jogos desenvolvidos para exercícios, reabilitação e recuperação envolvem movimentos repetitivos que podem deixar de ser atrativos para os usuários após algum tempo de jogo. Ainda que o indivíduo esteja jogando para seu próprio benefício, o jogo deve possuir fatores que incitem sua motivação para continuar jogando, como o nível de dificuldade, enredo, conexão com o jogador e nível de engajamento [45; 7; 20]. Além disso, algumas abordagens que utilizam jogos para monitoramento não levam em conta o fator idade [9]. Adolescentes e adultos jovens são um grupo de idade considerado complicado de obter participação ativa durante um tratamento [22]. Apesar de jogos serem indicados para prover motivação, eles perdem sua atratividade para a faixa etária à qual esse tipo de abordagem é direcionada.

O problema de falta de motivação é ainda mais evidente quando se fala no monitoramento de indivíduos saudáveis, que não têm como motivação direta o tratamento de uma doença já existente. Para estas pessoas, a atratividade do jogo em si é o único motivo pelo qual elas iriam utilizar o mesmo. Em resumo, enuncia-se o seguinte problema de pesquisa: como viabilizar o monitoramento de saúde de indivíduos utilizando jogos ainda mantendo a atratividade dos mesmos?

## 1.2 Objetivo

Neste trabalho, tem-se como objetivo desenvolver um arcabouço de software para tratar a aquisição de dados de diferentes sensores e dispositivos de entrada utilizados em jogos eletrônicos para posterior conversão destes dados em informação sobre a saúde do usuário. O arcabouço define componentes de software que encapsulam regras para a tradução dos dados do usuário em informação sobre a saúde. As regras definidas pelos componentes especificam informações sobre sintomas e características específicas, e classificam os dados capturados através da interação com os jogos, transformando-os em informação de saúde sobre o usuário.

Para permitir que os dados de saúde de um indivíduo sejam coletados durante o jogo de forma imperceptível para o indivíduo, a aquisição de dados deve ser feita por sensores e outros dispositivos de entrada de forma integrada com o enredo do próprio jogo. Os dados capturados são transmitidos para um componente do arcabouço, que os armazena e, posteriormente, os trata. Dessa forma, os eventuais movimentos específicos realizados pelo usuário que caracterizam deficiências psicomotoras, tais como tremores, dificuldades ou anomalias nos movimentos, serão registrados durante o curso do jogo, montando um histórico de saúde. Este histórico pode ser utilizado no futuro, por exemplo, para identificar o início de alguma deficiência de movimentos do indivíduo.

O arcabouço proposto possibilita a inserção de mecanismos de captura de dados de saúde a um jogo em tempo de desenvolvimento. Através de mecanismos bem definidos, são inseridos elementos no projeto do jogo que permitem que a captura de dados de saúde seja integrada de forma transparente ao seu fluxo. Esses elementos podem ser, movimentos específicos requeridos do jogador para controlar o jogo, ou um enredo e cenário adaptados para captar determinados dados em situações específicas, como, por exemplo, falar uma palavra específica para realizar análise de voz.

A hipótese é que jogos eletrônicos comuns, não direcionados para a saúde, continuam sendo atrativos ao suprirem a falta de motivação enfrentada pelos sistemas de monitoramento convencionais, tornando mais fácil a aquisição dos dados dos indivíduos durante o curso do jogo. Ao inserir mecanismos de aquisição de dados em gêneros de jogos comuns, tem-se uma maior possibilidade de aceitação do indivíduo, por não existir a pressão psicológica de

estar sendo monitorado.

Com o objetivo de realizar a validação, foram desenvolvidos dois jogos em parceria com o Instituto Federal de Alagoas (IFAL) utilizando a abordagem proposta para capturar os dados. Um dos jogos é o *Pinball World*, que utiliza o acelerômetro de um dispositivo Android para controlar uma esfera através de um labirinto. O jogo captura os dados de tremor do usuário durante o curso do jogo. O segundo jogo, o *Catch the Spheres*, capta dados de movimentos do corpo através do Kinect, testando o reflexo e velocidade de movimento do jogador ao capturar bolas que vêm em sua direção. Um experimento para demonstrar o funcionamento do arcabouço foi realizado. Dezoito pessoas foram selecionadas para jogar e solicitadas a responder um questionário sobre suas percepções em relação ao jogo. Os resultados indicam que é possível utilizar jogos casuais para o monitoramento. Os jogos desenvolvidos foram considerados divertidos e de fácil entendimento pela maioria (61%) dos participantes do experimento. Eles também declararam que incluiriam jogos casuais, não direcionados para monitoramento da saúde, em sua rotina diária. Parte do experimento realizado consistiu em pedir para os participantes simularem um movimento mais lento durante o jogo. Foi possível detectar uma redução na velocidade de movimento de alguns participantes, quando lhes foi pedido para capturarem as bolas mais lentamente no jogo *Catch the Spheres*. Não foi detectado um nível de tremor significativo através do jogo *Pinball World*.

### 1.3 Relevância

Sabe-se que existem vantagens em descobrir os primeiros sinais de condições adversas de saúde mais cedo, como redução de custos com saúde e a redução de riscos relacionados a doenças debilitantes [23]. Para prover um tratamento mais eficaz, os primeiros sinais que caracterizam a manifestação de uma doença precisam ser detectados antes que sintomas maiores apareçam [27]. Uma das formas de detectar sinais primários de uma condição de saúde é o monitoramento.

Existe a necessidade de uma abordagem de avaliação do indivíduo que auxilie o médico na aplicação do seu conhecimento e que diminua a dependência de informações providas pelo indivíduo. Se os dados necessários para a avaliação forem capturados automaticamente, pode-se aumentar a qualidade da informação capturada, através coleta de dados de saúde

integrada à rotina do indivíduo, e reduzir a carga de trabalho dos profissionais de saúde que dela necessitam. Capturar dados de saúde durante uma atividade prazerosa para o indivíduo como um jogo interativo é uma forma de monitoramento diferente das tradicionais, que torna o indivíduo mais propenso a contribuir com o seu monitoramento.

A principal contribuição deste trabalho é a disponibilização de um arcabouço para desenvolvimento e análise de dados direcionados para o monitoramento da saúde. No contexto de computação pervasiva, o arcabouço provê uma grande contribuição por permitir o monitoramento de diversos parâmetros de forma integrada à rotina do indivíduo. O desenvolvimento de jogos para monitoramento de dados de saúde e a análise dos dados obtidos são facilmente acessíveis para o desenvolvedor. O único passo necessário pro parte do desenvolvedor é escrever os dados coletados em um formato específico, para ter acesso a módulos e métodos de análise de dados providos pelo arcabouço.

Este trabalho avança o estado da arte das soluções de monitoramento não invasivo, contribuindo com atuais pesquisas dentro do laboratório Embedded, da Universidade Federal de Campina Grande, e também serve como base para futuras pesquisas na área de *Pervasive Healthcare*.

## 1.4 Organização

Este documento está estruturado da seguinte forma. No Capítulo 2, apresenta-se a fundamentação teórica, onde são apresentados os principais conceitos citados durante o trabalho, disponibilizando embasamento teórico para os leitores. No Capítulo 3, são apresentados os principais trabalhos realizados no domínio de monitoramento através de jogos. No Capítulo 4, apresenta-se o Arcabouço de Software para a Aquisição de Dados de Saúde Através de Jogos Eletrônicos. No Capítulo 5, são apresentados os dois estudos de caso construídos para validar o arcabouço. E, finalmente, no Capítulo 6 apresenta-se a conclusão do trabalho.

# Capítulo 2

## Fundamentação Teórica

### 2.1 Assistência Médica Pervasiva (*Pervasive Healthcare*)

O potencial de aplicação da computação pervasiva está presente em várias perspectivas e ambientes, como hospitais, situações de emergência, na indústria, educação, entre outros. O termo Assistência Médica Pervasiva, ou *Pervasive Healthcare*, é utilizado para descrever a integração da computação pervasiva na assistência médica. A necessidade de sistemas de assistência médica se deve ao fato de que profissionais da saúde precisam de mobilidade, além de terem que compartilhar suas experiências e informações com os demais profissionais responsáveis por determinado paciente [43; 2].

A assistência médica pervasiva é uma mudança de paradigma que, com o suporte apropriado, pode tornar as pessoas participantes ativas de seu próprio cuidado médico, prevenindo complicações e reduzindo o avanço da deterioração de suas condições. Para pacientes de alto risco, o cuidado proativo de profissionais de saúde seguindo protocolos, planos e objetivos requer o uso de uma estrutura de tecnologias de comunicação e informação, por exemplo, prontuários eletrônicos. A assistência médica pervasiva usa a infraestrutura de tecnologias de comunicação e informação para prover suporte a um estilo de vida assistido e independente, mantendo as pessoas em seu ambiente por tanto tempo quanto possível, evitando a necessidade de serem internados [10].

Dentre os possíveis benefícios do uso da Assistência Médica Pervasiva tem-se: a melhoria do tratamento domiciliar do paciente [1], a realização do monitoramento remoto e monitoramento contínuo em pacientes que possuem níveis cognitivos e físicos comprome-

tidos [11]. Entretanto, a concepção de um sistema de monitoramento contínuo e que não seja invasivo ainda é um grande desafio [4]. A necessidade de integrar diferentes sensores em uma única solução torna essa atividade mais difícil, devido a dispositivos pesados, visíveis e estereotipados, como um sensor de ECG, por exemplo. Por esse motivo esses dispositivos enfrentam resistência por parte dos usuários, que passam a não utilizá-los no dia a dia, perdendo a possibilidade de monitoramento contínuo da saúde [1; 4].

## 2.2 Jogos Sérios

Jogos sérios são jogos que foram projetados especificamente para treinamento e educação. A maioria das pessoas pensa em jogos como uma forma de entretenimento. Entretanto, há um interesse crescente em usar jogos para educar e treinar pessoas [5].

Apesar de existirem muitas definições para o termo, é consenso entre diferentes autores que o termo refere-se ao uso de jogos de computador cujo o principal propósito não é puramente entreter. De fato, jogos sérios têm sido aplicados em diversas áreas, como treinamento corporativo, cultural e militar, saúde e educação. Muitas dessas áreas estão relacionadas e se sobrepõem, como *e-learning* (educação através de meios eletrônicos), *edutainment* (entretenimento com propósitos educativos) e aprendizado baseado em jogos comuns e digitais [31]. O campo da medicina tem uma história de utilização de jogos como meios de engajar os pacientes de forma comportamental para melhorar os resultados na sua saúde. Há relatórios antigos de estudos de caso usando jogos com pacientes passando por doenças ou deficiências físicas [22].

Watters et al. [45] afirmam que jogos para a saúde são direcionados a aumentar a motivação de pacientes em três áreas. Primeiro, aumentar a motivação do paciente para aprender mais sobre sua condição e seu tratamento. Segundo, para ser usado como uma ferramenta em uma terapia de distração para dor e ansiedade. E, por último, para encorajar pacientes jovens a continuarem com seus tratamentos prolongados. Nesta última categoria, os jogos agem como "treinadores". Quando usados por crianças, eles podem ajudar reforçando as informações passadas na fase inicial do tratamento, provendo um sistema de lembrete, encorajando a prática de habilidades e gravando dados do tratamento e estado do usuário.

## 2.3 Exergames

*Exergame* é um termo formado a partir da combinação de *exercise* e *digital gaming*. É uma prática de exercícios físicos através de jogos eletrônicos [35]. Esse tipo de jogo envolve o jogador em atividades que o permitem exercitar-se para desenvolver habilidades motoras durante o jogo, focando em grupos musculares maiores em vez de destreza manual ou capacidades motoras finas [37]. Esse novo paradigma de entretenimento possibilitou à indústria de jogos criar títulos para favorecer a prática de exercícios físicos, visando melhorar o condicionamento físico dos usuários [39; 18]. Atualmente existem trabalhos na área de fisioterapia verificando a eficácia desses jogos para condicionamento físico, reabilitação e aprendizagem motora para pessoas com problemas de mobilidade.

Apesar dos *exergames* terem se tornado populares recentemente, a ideia de combinar movimento físico e seu reconhecimento através de visão computacional ou outra tecnologia sensorial não é inteiramente nova. Em 1982, o Atari Puffer<sup>1</sup> foi desenvolvido, mas não lançado, como o primeiro *exergame* usando uma bicicleta ergométrica para controlar jogos no Atari 2600.

Nos últimos anos, a tendência de *exergames* comerciais não é somente usar os videogames como instrumento motivacional para encorajar a prática de exercícios, mas também combiná-los com a coleta de parâmetros vitais. Um exemplo proeminente disso é o *EA Sports Active*, que tem como objetivo não somente reconhecer atividades via acelerômetros e medir a pulsação para propósitos relacionados à saúde (telemonitoramento e análise de dados de exercícios), mas também melhorar o treinamento e resultados do uso de *exergames* em geral [18].

Os jogos *Wii Sports* (2006) e *Wii Fit* (2008) (Figura 2.1), produzidos pela Nintendo, são combinações de jogos e prática de exercícios, o que os insere na categoria de *exergames*. Medindo pela popularidade dos jogos, pode-se simplesmente classificá-los como jogos unicamente lúdicos. A diferença é que o *Wii Sports* foca mais em ser um jogo lúdico, mas o *Wii Fit* também tem como objetivo ensinar algo sobre prática de exercícios. O principal fator de efeito dos exercícios no *Wii Sports* é que ele utiliza controladores com sensores de movimento, como o *Wii Remote* (Wiimote) e o *Nunchuk*, que força o jogador a mover-se

<sup>1</sup><http://kotaku.com/5009577/atari-puffer-the-wii-fit-of-1982>

fisicamente quando está jogando. O Nunchuk é uma extensão que pode ser conectada ao Wiimote para possibilitar o controle de jogos com as duas mãos. No Wii Fit, uma balança também é utilizada durante o jogo [38].



Figura 2.1: Os jogos Wii Fit e Wii Sports

## 2.4 Sensores para Captura de Dados de Saúde

Através de determinados sensores para captura de dados de saúde, é possível capturar uma ou mais características que podem auxiliar no diagnóstico e monitoramento de doenças. Redes de sensores montadas no lar podem ajudar pessoas e seus cuidadores, provendo monitoramento médico contínuo, controle de aparelhos domésticos, acesso a dados médicos e comunicação de emergência. O monitoramento do estado de saúde das pessoas é o tipo de aplicação mais estudado em sistemas de assistência médica pervasiva. Os sinais vitais mais utilizados são eletrocardiograma (ECG), oximetria de pulso, temperatura corporal, batimentos cardíacos e pressão sanguínea. Os dados de aceleração também são utilizados com esses sinais vitais em alguns estudos [4]. No contexto de jogos, a utilização de alguns sensores é impraticável, devido a seu tamanho, peso e forma de utilizar. Entretanto, sensores como câmeras [36; 33], acelerômetros e giroscópios[37], medidor de pulsação [25] e Interfaces Cérebro-Computador [34] vêm sendo utilizados com sucesso em jogos:

- **Acelerômetros e giroscópios:** um típico acelerômetro com seis graus de liberdade detecta aceleração em três eixos espaciais e a rotação ao redor desses eixos. Acelerômetros são usados no Wiimote e Nunchuck da Nintendo e em outros tipos de jogos

para detectar e interpretar diferentes tipos de movimento. A acurácia do Nintendo Wii Motion Plus é melhorada utilizando giroscópios.

- **Câmeras:** podem ser usadas para capturar as posições e movimentos dos jogadores. Sistemas de visão básicos incluem o Sony EyeToy e o PlayStation Eye. Vários jogos acadêmicos usam uma ou mais câmeras para capturar o movimento do jogador. Uma abordagem relacionada é utilizar uma câmera para rastrear marcações infravermelhas. A visão computacional pode ser aumentada com outras tecnologias para melhorar sua acurácia, como, por exemplo, um sensor de profundidade utilizado para produzir imagens 3D no Kinect, da Microsoft.
- **Medidores de pulsação:** são usados para promover a execução eficiente de atividades físicas, ajustando a intensidade dos exercícios baseado nos batimentos cardíacos. O grau de dificuldade precisa ser ajustado efetivamente, para que os exercícios não se tornem monótonos.
- **Interfaces Cérebro-Computador:** lêem e analisam a atividade cerebral do jogador em tempo real. Os dois principais hardwares disponíveis no mercado são o Emotiv EPOC<sup>2</sup> e o MindSet, da NeuroSky<sup>3</sup>. Os dispositivos traduzem diretamente as intenções do jogador em ações dentro do jogo, dispensando dispositivos como mouse, teclado ou joystick.

---

<sup>2</sup><http://www.emotiv.com>

<sup>3</sup><http://www.neurosky.com>

## Capítulo 3

### Trabalhos Relacionados

Apesar de se pensar em jogos apenas como uma forma de entretenimento, existe um interesse crescente em utilizá-los para outras finalidades, devido a seu caráter lúdico. Jogos eletrônicos têm sido usados para vários propósitos. No âmbito educacional, os jogos ajudam os estudantes no seu aprendizado, provendo uma forma diferente, motivadora e eficiente de absorver conteúdo. Simuladores, que também podem ser encaixados na definição de jogos educacionais, oferecem uma alternativa viável para o treinamento de pessoas. No campo de assistência à saúde, existem jogos comerciais que têm o propósito de atingir certos objetivos comportamentais [22]. Além dos propósitos já mencionados, muitos trabalhos tratam sobre jogos que cuidam da motivação de pacientes para fazer exercícios e reabilitação de pacientes com deficiências motoras.

#### 3.1 Video Games in Health Care: Closing the Gap

Kato [22] apresenta outros efeitos positivos de *video games*, especialmente aqueles com foco em assistência à saúde. Segundo a autora, seres humanos nem sempre se comportam de forma a tomar vantagem do que a assistência à saúde tem a oferecer. A maioria das pessoas não age em conformidade com as condições que podem salvar suas vidas. As soluções para esse problema são claramente complexas, mas os fatores psicológicos e comportamentais têm um papel fundamental nessas soluções. Neste sentido, jogos vêm sendo utilizados cada vez mais para endereçar as barreiras psicológicas e comportamentais para a assistência à saúde ótima.

O mecanismo de ação principal dos jogos geralmente citado é sua capacidade de aumentar a motivação. O foco da atenção em uma distração provida por um jogo é visto como um fator fundamental para explicar como indivíduos usam jogos para lidar com sintomas adversos, comportamentos dolorosos, adversos e entediantes.

## 3.2 Adaptive Virtual Reality Games for Rehabilitation of Motor Disorders

Ma et al. [24] apresenta um sistema de treinamento para encorajar pacientes a praticarem exercícios físicos. O trabalho apresenta diretrizes úteis para a construção de um arcabouço para jogos interativos direcionados à saúde. Um dos fatores citados, que deve ser levado em conta na definição de um arcabouço, é a construção de um perfil do paciente. De acordo com os autores, um fisioterapeuta geralmente criará exercícios de terapia motora para um paciente baseado em um número de fatores como: idade, sexo, histórico cultural, mão com que escreve e condição médica (por exemplo, tempo desde o último derrame, hemiplegia do lado direito ou esquerdo e habilidades cognitivas, sensoriais e motoras, baseadas em testes padronizados).

Utilizando os dados do perfil e também alguma informação de contexto, não somente o nível de dificuldade dos exercícios, como também as partes do corpo ativas durante a tarefa poderão ser adaptadas às necessidades do paciente. As tarefas podem ser desenvolvidas para treinar um movimento específico, como extensão do punho, para aumentar a extensão do movimento ou a resistência da junta do punho.

A parte do sistema que trata da adaptação dinâmica usa os dados do perfil do paciente e os dados de seu progresso para selecionar atividades e configurar o nível de dificuldade das tarefas e do jogo. Assim, o sistema pode prover dados sobre o progresso individual dos pacientes, que podem ser comparados durante o período de recuperação para montar uma base de dados histórica do progresso do paciente. Um módulo de análise de dados possibilita a visualização das trajetórias de movimento dos pacientes e mostra os ângulos das juntas, extensão dos movimentos e velocidades.

Na Figura 3.1, ilustra-se a arquitetura do sistema. A entrada do ambiente de realidade virtual para treinamento se dá através de mouse e teclado para operadores do sistema,

e dispositivos de captura de movimento em tempo real: duas luvas *5DT Ultra DataGloves*<sup>1</sup> (Figura 3.2c), além de quatro sensores magnéticos sem-fio *Ascension MotionStar*<sup>2</sup> (Figura 3.2a), que são usados para captar movimentos das mãos, braços e parte superior do corpo. A saída envolve modalidades visuais, sonoras e hápticas. A interface de saída visual inclui um monitor para operadores e um dispositivo com visor montado sobre a cabeça (*Head Mounted Display - HMD*) de alta resolução (Figura 3.2b).

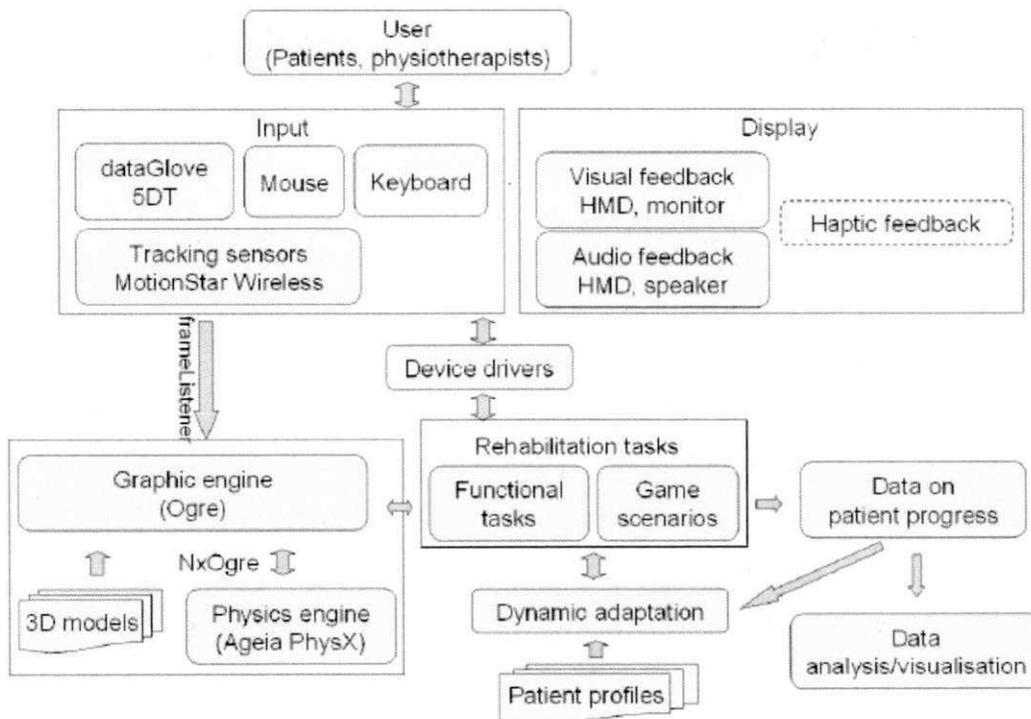


Figura 3.1: Arquitetura do sistema de realidade virtual para reabilitação física [24]

### 3.3 Context Aware Serious Games Framework for Sport and Health

O *SocialAware* [20] é um arcabouço para a construção de serviços cientes de contexto. No trabalho, é apresentado um conjunto de jogos sérios para suporte a esporte e saúde que, através do uso do arcabouço, são transformados em jogos cientes de contexto. A captura do

<sup>1</sup><http://www.5dt.com/products/pdataglove5u.html>

<sup>2</sup><http://www.ascension-tech.com/realtime/RTMotionSTARTethered.php>

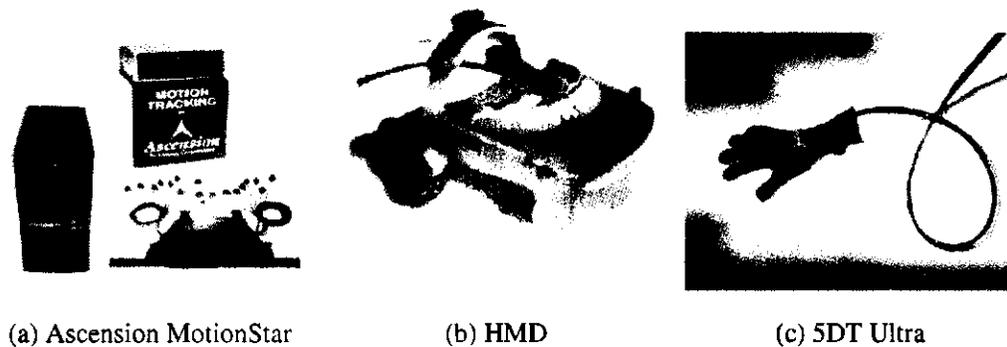


Figura 3.2: Dispositivos de entrada

contexto do usuário, provendo serviços de *exergames* apropriados é vista como um auxílio para solucionar muitos problemas de saúde. Apesar de a maioria dos problemas de saúde ocorrerem principalmente devido à falta de atividade física, muitas são as pessoas que têm motivação para praticar tais atividades. Os jogos sérios para suporte a esportes e saúde motivam as pessoas a praticarem exercícios durante a atividade do jogo, ou as ensinam sobre problemas de saúde.

Na Figura 3.3a, ilustra-se a arquitetura do sistema e na Figura 3.3b, ilustra-se a interface de visualização do *SocialAware* mostrando serviços diferentes em dois contextos distintos. Foram usados dados de aceleração, temperatura e altura para calcular um nível de estresse e, dependendo do valor calculado, são sugeridos jogos diferentes. A captura de tela no topo mostra a interface durante o contexto *no trabalho*. Propõe-se que o usuário jogue um jogo após ler um email importante, por exemplo. A tela de baixo mostra uma situação em casa. Os autores mencionam que não desejam que seus jogos sejam vistos como substitutos de esportes reais. Dessa forma, usando dados da previsão do tempo e hora, pode-se sugerir atividades ao ar livre, se possível. Se o sistema identificar que o usuário não realizou atividade aeróbica no dia anterior, por exemplo, pode sugerir uma caminhada ou um *exergame* aeróbico.

Estudos multidisciplinares recentes no domínio de sensores sem-fio, smartphones, redes sociais e comunicação móvel contribuíram para a computação ciente de contexto. Dessa forma, é possível capturar diferentes informações de contexto relacionadas a sinais vitais, como batimento cardíaco, pressão sanguínea, nível de glicose, condições do suor, etc. Além

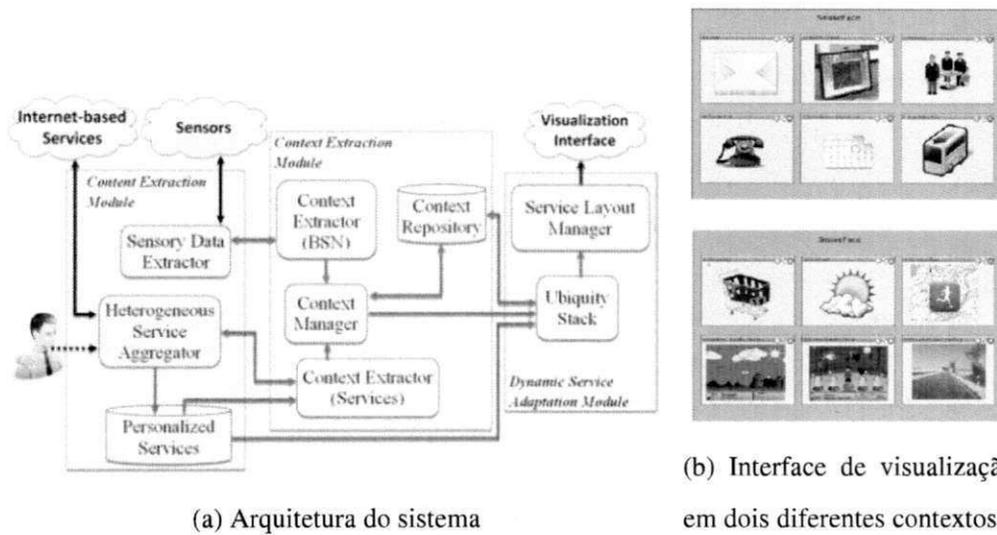


Figura 3.3: SocialAware

disso, podem ser capturadas ações, como andar, dormir, dirigir, cair, correr, falar e conversar com um amigo, e também variáveis de ambiente, como temperatura, umidade do ar, localização, altitude, etc.

Extrair informações de contexto a partir de dados de sensores e informação multimídia contida em serviços heterogêneos traz dois desafios. Primeiramente, é necessária a extração em tempo real de conteúdo multimídia e de redes de sensores corporais para análise de contexto. Depois, é preciso aplicar a lógica apropriada para extrair a informação de contexto das redes sociais e dados de sensores, agregá-la e então armazená-la em um repositório para ser usada em uma seleção dinâmica de jogos sérios e serviços de saúde relevantes.

Três módulos lógicos são descritos para melhor representar o conceito:

1. *Módulo de Extração de Conteúdo*: responsável por extrair as informações de serviços heterogêneos de Internet e da rede de sensores corporais.
2. *Módulo de Extração de Contexto*: encontra o valor contextual de um serviço. Extratores de conteúdo recebem conteúdo em tempo real de sensores e serviços de Internet e passam o conteúdo e os metadados de cada serviço para os componentes apropriados.
3. *Módulo de Adaptação Dinâmica de Serviço*: mapeia dinamicamente um subconjunto de jogos sérios e/ou serviços relacionados à saúde extraídos de uma base de dados

de Serviços Personalizados, selecionando serviços, jogos e níveis de exercício únicos para cada pessoa.

Para avaliar a efetividade do sistema proposto, foi conduzido um teste de avaliação subjetiva entre maio e novembro de 2010, baseado em uma amostra de 95 pessoas, das quais 25% são mulheres. A maior parte dos usuários tinham mais de 25 anos e 45 deles tinham menos de 25 anos. Todos os participantes utilizaram diversos serviços de Internet em diferentes escalas nos últimos 4 a 6 anos. A ideia de utilizar sugestões de serviços dinâmicas foi fortemente aceita por 65% dos usuários. Além disso, alguns usuários levantaram a questão da privacidade, enquanto outros usuários preocuparam-se com o preço dos sensores. Alguns demonstraram pessimismo devido à necessidade de sempre carregar os sensores para obter os serviços.

### **3.4 Serious Game Based on First Aid Education for Individuals with Autism Spectrum Disorder (ASD) Using Android Mobile Services**

Em [14] é apresentado um conjunto de jogos para ajudar pessoas com Desordem do Espectro Autista (DEA). O paciente é motivado a jogar durante o seu progresso e atividades executadas durante o jogo são guardadas para gerar relatórios, que serão acompanhados pelo paciente, sua família e/ou médicos. Primeiramente, um conjunto de mini-jogos é apresentado ao usuário, cada um com diferentes níveis de dificuldade. Após terminar a atividade, a aplicação conecta-se com o servidor, acessa a base de dados e armazena os dados gerados durante o jogo, atualizando os registros daquele usuário, se necessário. Com os dados salvos, é gerado um relatório detalhado sobre o usuário, explicando em qual atividade foram cometidos erros e qual a possível razão.

### 3.5 Game based Learning to Enhance Capabilities of Elderly People

Arntzen [6] discute os conceitos, realidade sócio-econômica e requisitos técnicos para desenvolver jogos para idosos. Sabe-se que tecnologias de Realidade Virtual (RV) têm o potencial de contribuir na melhoria de sessões de terapia física, tratamento do mal de Alzheimer, reabilitação de infarto e assim por diante. O propósito do trabalho é desenvolver um jogo que contribua para melhorar as habilidades cognitivas e físicas de idosos ou deficientes. Desde o lançamento do Nintendo Wii em 2008, os sensores de movimento nos jogos possibilitaram o conceito de *exergames*. Além disso, novos jogos surgiram, provando a sua capacidade de melhorar as habilidades cognitivas através do estímulo das atividades mentais.

Estudos recentes indicam que as pessoas podem evitar o declínio de habilidades cognitivas ou físicas se forem motivadas a exercitar seu cérebro e corpo. Os novos sistemas de jogos têm sido vistos com potencial por fisioterapeutas e outros profissionais de saúde. Entretanto, desenvolver novos jogos requer entender as habilidades cognitivas e físicas dos idosos que geralmente diminuem com o tempo e, também, é necessário uma preocupação em relação à limitação física dos jogadores de idade avançada ou deficientes.

Ainda em [6], no projeto de jogo *SmartBrain* aplicado à doença de Alzheimer, são utilizados meios para avaliar a escuta, reconhecimento de cores, avaliação da memória e execução de movimentos. O usuário não deve correr riscos de machucar-se devido a movimentos bruscos durante o jogo. É crucial que o jogador sintam-se seguros ao jogar.

### 3.6 Conclusão

Nesta seção foram apresentados os principais trabalhos que utilizam jogos com algum propósito relacionado à saúde. As soluções propostas são geralmente invasivas e, apesar terem como intenção serem utilizadas pelos usuários por um longo período, não existe a preocupação em manter o enredo do jogo atrativo para o usuário e criar nele a motivação para continuar jogando. A diferença para o trabalho aqui desenvolvido é o direcionamento para pessoas saudáveis e a preocupação em manter o projeto de jogo atrativo, embutindo nele os métodos de monitoramento.

## Capítulo 4

# Arcabouço para a Aquisição de Dados de Saúde Utilizando Jogos Eletrônicos

Este capítulo apresenta o arcabouço que permite o desenvolvimento de jogos eletrônicos para a aquisição de dados de saúde. São apresentados os requisitos funcionais do arcabouço e, em seguida, é feita uma descrição detalhada da arquitetura do arcabouço, mostrando seus componentes e funcionalidades.

### 4.1 Requisitos do Arcabouço

Para possibilitar o desenvolvimento de jogos para captura de dados de saúde, um arcabouço deve permitir a aquisição e persistência dos dados obtidos a partir da interação do usuário com sensores. A grande quantidade de dados coletada deve ser gerenciada pelo arcabouço de monitoramento de saúde, apoio à decisão e processamento de informação clínica para auxiliar médicos em suas decisões e melhorar o entendimento da dinâmica da evolução de doenças. O conhecimento médico é frequentemente atualizado e reavaliado para incluir a identificação de novos fatores de risco e novas evidências de estudos clínicos. Os desafios enfrentados atualmente são: incorporar em sistemas pessoais de monitoramento de saúde o conhecimento mais recente e baseado em evidências; e transformar os dados coletados em conhecimento para suportar o processo de tomada de decisão. A tecnologia pode ter um papel fundamental ao focar em uma abordagem baseada em conhecimento, como identificação de fatores de risco, testes diagnósticos e novas evidências de estudos clínicos, para integrar

informações passadas e atuais de cada indivíduo, junto com evidências estatísticas [40].

Um requisito do arcabouço é permitir que a coleta dos dados seja realizada de forma imperceptível para o indivíduo, embutida no enredo do próprio jogo no qual é utilizado. Outras informações relevantes, como idade, gênero e medidas antropométricas, como peso e altura, podem ser obtidas em posteriormente, com o conhecimento do usuário, quando se fizer uso dos dados de saúde coletados pelo arcabouço.

## 4.2 Arquitetura

Na Figura 4.2, apresenta-se uma visão geral da arquitetura da solução, que é constituída por quatro componentes: *Web Service*, *Gerenciador de Dados*, *Analizador de Dados* e o *Banco de Dados*. A estrutura estática do arcabouço pode ser vista no diagrama de classes, na Figura 4.3. Na Figura 4.1, apresenta-se a transferência de dados entre as classes do arcabouço.

O processo se inicia com a aquisição dos dados dos sensores (1), 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` (2), também acessado através do `DataManager`, para armazená-los no *banco de dados*. A partir daí, os dados podem ou não ser filtrados através do `FilterModule` pelas classes de filtro que implementam a interface `IFilter`, e depois enviados para o `DataAnalyzer` (3). No `DataAnalyzer`, os dados estão prontos para serem processados pelas classes de análise que implementam a interface `IAnalyzer`. O `RuleManager` utiliza o `FileManager` para ler as regras de um arquivo (4) e as transfere para o `DataAnalyzer` (5), assim, os dados podem ser interpretados. O `DataAnalyzer` pode também enviar os dados para o `WriterModule` (6), que os escreverá em arquivo para visualização em outros programas, ou utilização em métodos de aprendizado de máquina, por exemplo.

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 *web service*. O formato aceito pelas operações é o JSON (JavaScript

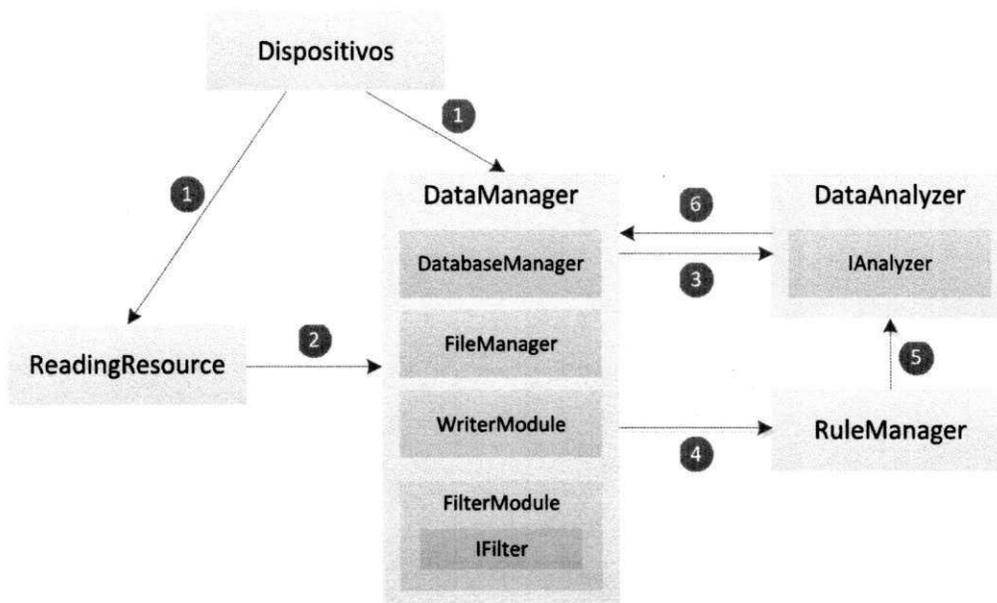


Figura 4.1: Transferência de Dados Dentro do Arcabouço

Object Notation). Na Tabela 4.1, ilustram-se as operações disponibilizadas pelo *web service* e um exemplo de como os dados devem ser estruturados para cada operação.

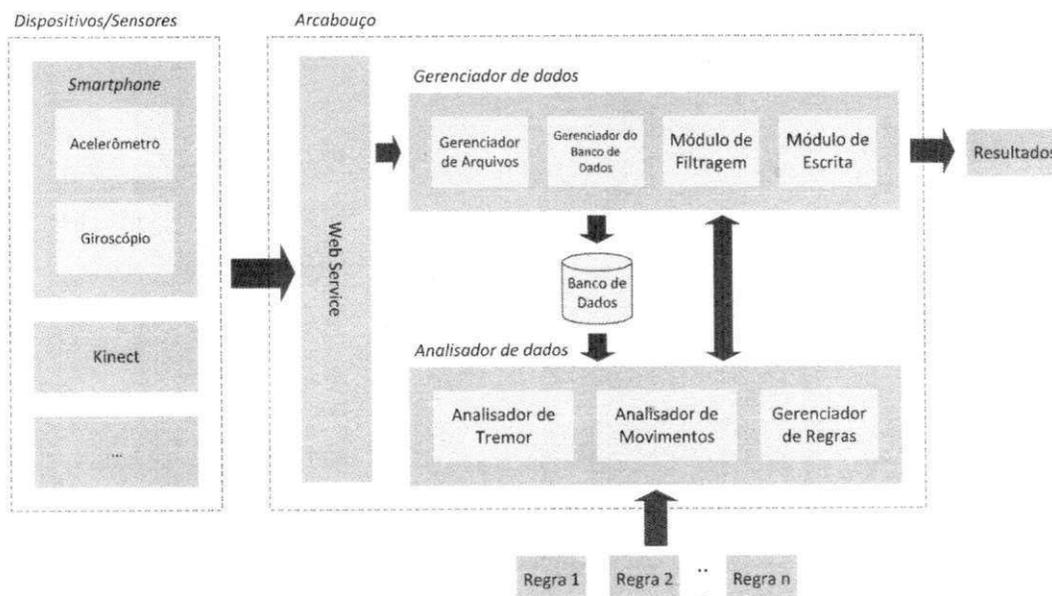


Figura 4.2: Arquitetura do Arcabouço

Tabela 4.1: 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>

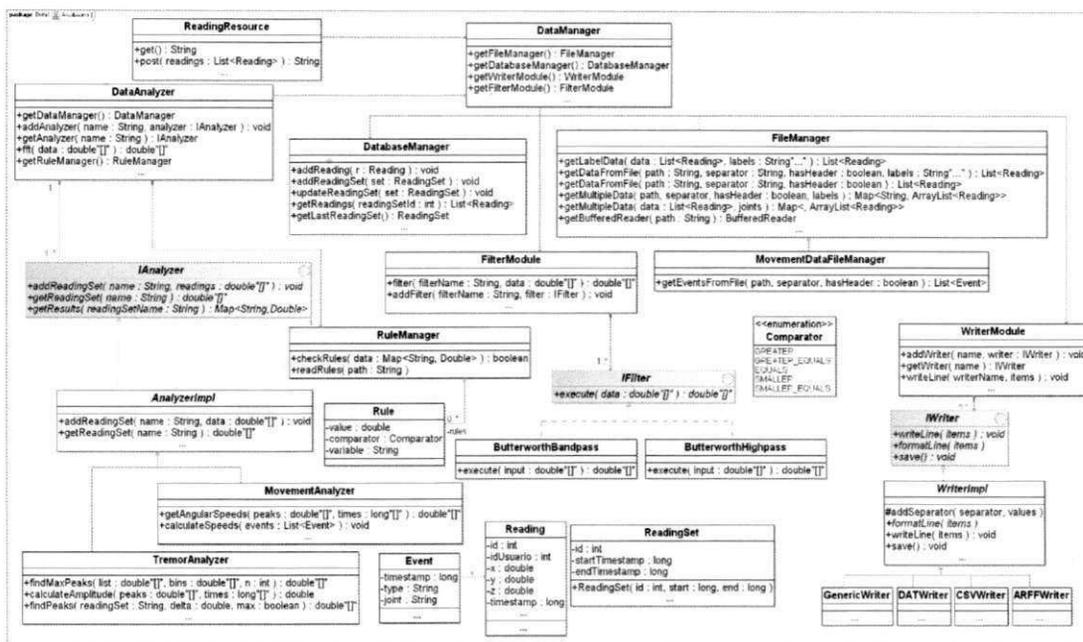


Figura 4.3: Diagrama de Classes do Arcabouço

### 4.2.1 Módulo Gerenciador de Dados

O *Gerenciador de Dados* possui submódulos responsáveis por fazer leitura, separação e filtragem dos dados, além do gerenciamento destes no Banco de Dados e escrita dos resultados disponibilizados pelo *Analizador de Dados*. 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*, que são 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 utilizando a biblioteca *Jersey*<sup>1</sup>, que facilita o desenvolvimento de *RESTful webservices*. As requisições são enviadas para serem processadas pela

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

classe `ReadingResource`, que é um *web resource*, uma entidade que recebe requisições HTTP e envia respostas. Esta classe possui dois métodos, o `get()` 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 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*, e 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`.
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

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

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

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 3, 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.

### Módulo de Escrita

O *Módulo de Escrita* é implementado pela classe `WriterModule`, e 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 1.
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`.

### Módulo de Filtragem

O *Módulo de Filtragem* faz o pré-processamento dos dados, selecionando apenas os espectros de frequência de interesse. A classe `FilterModule` implementa as funcionalidades do *Módulo de Filtragem*. Ela permite o gerenciamento de filtros, implementados em classes distintas, indexados por um nome e acessados por métodos explicados a seguir.

1. `addFilter(String filterName, IFilter filter)` adiciona um novo filtro `filter` indexado pelo nome `filterName`.
2. `filter(String filterName, double[] data)` executa o filtro de nome `filterName` sobre os dados `data` e retorna um array `double[]` de mesmo tamanho que `data` com os dados filtrados.

A interface `IFilter` define que as classes que implementarem filtros, deverão receber os dados através do método `execute(double[] data)`, e retorná-los, depois de processados, dentro de um array `double[]`. Atualmente, duas classes de filtros estão implementadas: `ButterworthHighpass`, um filtro passa-alta de frequência de corte de 1Hz e `ButterworthBandPass`, um filtro passa-faixa de frequências de corte de 2Hz e 8Hz, suficientes para identificar os tremores que ocorrem nas mãos.

Filtros passa-alta com frequência de corte de 1Hz são geralmente utilizados em aplicações de análise de movimentos usando acelerômetros para remover mudanças na orientação de segmentos do corpo, ajustes de postura e os efeitos da gravidade. Os componentes de frequência mais alta (2-16Hz) refletem as acelerações de segmentos do corpo, tipicamente

associadas com movimentos rápidos, marcados por fases de aceleração/desaceleração [12; 29].

### Gerenciador do Banco de Dados

O *Gerenciador do Banco de Dados*, implementado na classe `DatabaseManager` dá acesso ao mecanismo de persistência do arcabouço, que fornece operações de armazenamento e leitura dos dados coletados. A implementação do mecanismo de persistência do arcabouço baseia-se no Sistema de Gerenciamento de Banco de Dados (SGBD) Relacional MySQL<sup>6</sup>. A comunicação entre o *Gerenciador de Dados* e o banco de dados MySQL é feita através do Hibernate<sup>7</sup>, um *framework* para mapeamento objeto-relacional escrito na linguagem Java, permitindo consultas em SQL (*Structured Query Language*) ou HQL (*Hibernate Query Language*).

A classe `DatabaseManager` disponibiliza os seguintes métodos para manipulação dos dados:

1. `getLastReadingSet ()` Retorna o identificador *id* do último `ReadingSet` gravado no banco de dados incrementado de 1. As novas leituras são adicionadas ao `ReadingSet` de identificador *id* + 1. Isso permite adicionar um grande número de leituras por etapas. Enquanto `getLastReadingSet ()` não for chamado novamente, todas as leituras serão adicionadas ao `ReadingSet` que for retornado pelo método.
2. `addReading (Reading r)` adiciona uma nova leitura *r*.
3. `addReadingSet (ReadingSet set)` adiciona um novo conjunto de leituras *set*.
4. `updateReadingSet (ReadingSet set)` atualiza o conjunto de leituras *set* passado como atributo com suas novas informações. O identificador do *set* deve estar inicializado, para que este seja localizado no banco de dados.

<sup>6</sup>Disponível em: <http://dev.mysql.com/downloads/>

<sup>7</sup>Disponível em: <http://www.hibernate.org/downloads>

5. `getReadings(int readingSetId)` retorna uma `List<Reading>` com as leituras pertencentes ao conjunto de leituras de identificador `readingSetId`.

### 4.2.2 Analisador de Dados

O *Analisador de Dados*, implementado pela classe `DataAnalyzer`, realiza o processamento dos dados e retorna as informações necessárias para a análise. A classe `IAnalyzer` especifica os métodos que uma classe de análise de dados deve possuir. Uma nova classe deve implementar a interface `IAnalyzer`. Dois tipos de análise são suportadas atualmente, a análise de tremor e análise de movimentos, realizadas respectivamente pelo *Analisador de Tremor* e *Analisador de Movimentos*. A análise é feita baseada em regras providas ao sistema, utilizadas para definir como classificar as informações.

Os métodos que uma nova classe de análise deve possuir são os seguintes:

1. `addReadingSet(String name, double[] readings)` adiciona o novo conjunto de leituras `readings` sob o nome `name`.
2. `getReadingSet(String name)` retorna o conjunto de leituras sob o nome `name`.
3. `getResults(String readingSetName)` realiza a análise sobre o conjunto de leituras de nome `readingSetName` e retorna os resultados em um mapa `Map<String, Double>` onde a *chave* é o nome de uma variável de retorno da análise, e o *valor* é o valor da variável. A classe de análise deve apresentar todos os seus resultados dentro deste mapa, para avaliação pelo *Gerenciador de Regras*.

Os métodos disponibilizados pela classe `DataAnalyzer` são:

1. `getDataManager()` retorna a instância do *Gerenciador de Dados*.
2. `getRuleManager()` retorna a instância do *Gerenciador de Regras*.
3. `addAnalyzer(String name, IAnalyzer analyzer)` adiciona uma nova classe de análise ao *Analisador de Dados*, sob o nome `name` especificado.
4. `getAnalyzer(String name)` retorna o analisador de nome `name`.

5. `fft(double[] data)` executa a Transformada Rápida de Fourier (FFT) sobre os dados passados no array `data`. Retorna o resultado em um *array* `double[]`.

### Analizador de Tremor

A implementação deste módulo é feita pela classe `TremorAnalyzer`, que processa o sinal do acelerômetro, realizando a análise de espectro para identificar as frequências de tremor predominantes. Pequenos acelerômetros são usados em muitas aplicações clínicas, sendo bastante populares em aplicações para medição de tremor, e são capazes de medir acelerações menores que  $0,02g$  ( $1g = 9.807m/s^2$ , a aceleração estática da gravidade) [16]. O sinal proveniente do acelerômetro de três eixos é composto por coordenadas  $(x, y, z)$ , que representam a quantificação da força da gravidade sobre o eixo em um determinado momento. Na Figura 4.4, ilustra-se um esquema de um acelerômetro e seus eixos  $x, y$  e  $z$ , considerando o eixo  $z$  perpendicular à força da gravidade  $g$ .

Para realizar a análise espectral do sinal do acelerômetro, é utilizada a Transformada Rápida de Fourier (*FFT - Fast Fourier Transform*), acessada através do método `fft(double[] data)`, disponível na classe `DataAnalyzer`. A FFT transforma um sinal no domínio do tempo para o domínio de frequência. A maioria das análises espectrais é baseada na FFT, e sua simplicidade computacional permite uma implementação eficiente e análise de dados rápida. A análise espectral é um popular método para quantificar tremores, dada a característica oscilatória destes. A ideia é calcular a função de densidade espectral de energia (*Power Spectral Density - PSD*) em frequências diferentes por todo o espectro. A frequência dominante do tremor é evidente no maior pico na densidade espectral de energia [32].

Além dos métodos especificados pela interface `IAnalyzer`, os seguintes métodos são implementados na classe `TremorAnalyzer`:

1. `findPeaks(String readingSet, double delta, boolean max)`  
encontra os picos mínimos ou máximos no sinal contido no conjunto de leituras de nome `readingSet`. O parâmetro `delta` é a diferença máxima entre um pico local e geral, usado para encontrar os picos gerais. O valor de `delta` afeta o número de picos encontrados. O parâmetro `max` define se o método deve retornar os picos máximos (se `max` for `true`) ou máximos e mínimos (se `max` for `false`).

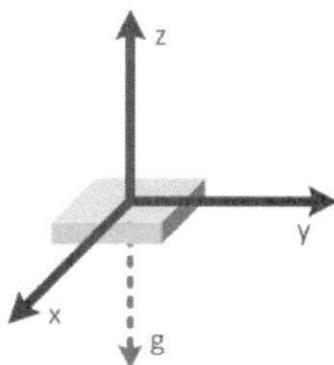


Figura 4.4: Eixos do acelerômetro

2. `calculateAmplitude(double[] peaks, double[] times)` retorna a amplitude em centímetros de um sinal de acelerômetro. A precisão do método depende da precisão do acelerômetro.

### Analizador de Movimentos

O módulo *Analizador de Movimentos* processa os dados obtidos do sensor de movimentos Kinect e medidas angulares obtidas de acelerômetros e giroscópios para calcular a velocidade angular e ângulo de inclinação de segmentos do corpo. O Kinect é um sensor de movimento que funciona baseado em uma câmera e um sensor de profundidade, captando coordenadas 3D de pontos com precisão. Sua eficácia na análise do movimento do corpo humano é comprovada, podendo substituir ferramentas de análise tridimensional quando uma alta acurácia não é necessária [41; 30]. Aplicações vão desde a avaliação do controle postural [13], à análise de marcha [17].

Esse módulo é capaz de calcular a velocidade linear de movimento de pontos do corpo a partir dos dados do Kinect. Esses dados consistem de coordenadas espaciais  $(x, y, z)$ , que são as posições do esqueleto capturadas pelo Kinect, cujos valores são dados em unidades métricas. Na Figura 4.5 é mostrado o *Skeleton Space*, um sistema de coordenadas orientado para a direita, que posiciona o Kinect na origem, com o eixo  $z$  se estendendo na direção para qual o Kinect aponta. O eixo  $y$  positivo se estende para cima e o eixo  $x$  positivo se estende para direita.<sup>8</sup> A Figura 4.6 mostra a representação gráfica das coordenadas da mão esquerda ao acenar, obtidas do Kinect durante aproximadamente 10 segundos.

<sup>8</sup>Disponível em: <http://msdn.microsoft.com/pt-br/library/hh973078.aspx>

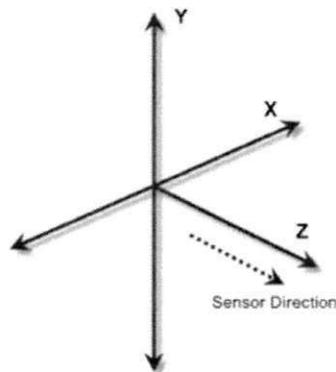
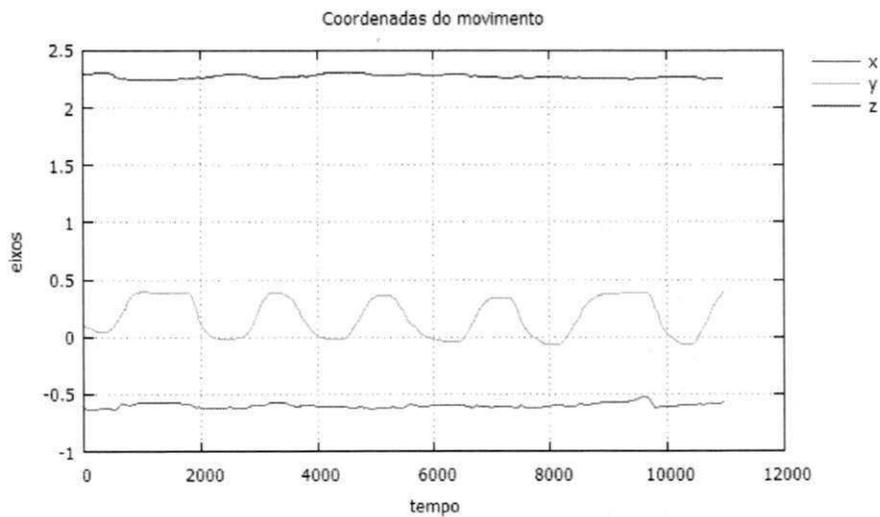
Figura 4.5: O *Skeleton Space*

Figura 4.6: Plotagem das coordenadas da mão esquerda

### Gerenciador de Regras

O *Gerenciador de Regras* é implementado pela classe `RuleManager`, e permite especificar regras para classificar os dados provenientes do *Analizador de Dados*. As regras permitem especificar condicionais que definirão uma interpretação para os dados. As regras são especificadas através de arquivos, uma regra por linha, no formato `variável sinal valor`, onde `variável` é uma palavra, sem espaços; `sinal` é um dos operadores {`<`, `<=`, `==`, `>=`, `>`} e `valor` é um decimal ou inteiro. No Código 4.1 é mostrado um exemplo de como estruturar o arquivo de regras.

---

Código Fonte 4.1: Um exemplo de formato do arquivo de regras

---

```
amplitude > 0
amplitude < 5
frequency >= 1
frequency < 9
```

---

As regras permitem verificar os resultados providos pelas classes analisadoras. Através das regras, por exemplo, podem ser providos intervalos normais de valores, e os valores obtidos do interação do usuário com os sensores são verificados para checar se são normais ou anormais. No caso de resultados anormais, é necessária uma investigação mais profunda para sua causa.

Os seguintes métodos fazem parte da classe `RuleManager`:

1. `readRules(String path)` carrega as regras contidas no arquivo provido em `path`. Este método deve ser chamado antes do método 2, ou a verificação das regras não será efetuada.
2. `checkRules(Map<String, Double> data)` verifica se os dados providos em `data` obedecem às regras definidas. O mapa `data` é provido por uma das classes de análise de dados.

## 4.3 Utilizando e Estendendo o Arcabouço

Nesta seção, encontram-se os passos a serem seguidos para utilizar o arcabouço e também estendê-lo. Por utilizar requisições HTTP e JSON para envio e representação dos dados, respectivamente, os métodos para coleta de dados durante o jogo são independentes de linguagem. Entretanto, o arcabouço foi feito em Java, e é necessário utilizar esta linguagem para usar as funções e partes do arcabouço.

### 4.3.1 Lendo e Armazenando os Dados de um Sensor Durante um Jogo

A coleta dos dados deve ocorrer em uma frequência apropriada para captar atualizações significativas nos valores obtidos dos sensores. Para dados obtidos através de acelerômetros, por exemplo, é recomendável escolher a frequência de amostragem maior que a frequência de ocorrência do evento que se deseja monitorar. Para identificar tremores, a frequência de

amostragem deve ser pelo menos de 14 Hz para captar todos tipos de tremores que ocorrem nos membros (ver Tabela 5.1). Outros dados, como pulsação, podem ser lidos a uma taxa de amostragem menor, por não possuírem uma alta taxa de atualização.

Para taxas de amostragem altas, recomenda-se que o envio para o *webservice* seja feito de 1000 em 1000 leituras, aproximadamente, ou que todas as leituras sejam armazenadas em arquivo, para serem processadas posteriormente pelo arcabouço. Para taxas de amostragem baixas, as leituras podem ser enviadas ao fim de uma sessão de alguns minutos de jogo. O Código 4.3 mostra um exemplo em Java de como as leituras devem ser armazenadas e enviadas.

O método `enviar()` pode deter a execução do código por algum tempo, devido à operação de enviar os dados pela rede. Para que nenhuma leitura do sensor seja perdida, no caso de uma taxa de amostragem alta, o ideal é que as operações dentro de `enviar()` sejam executadas em uma *thread* diferente.

O objeto JSON *leitura*, mostrado no Código 4.2 que representa uma leitura do sensor deve conter os atributos *id*, *idUsuario*, *x*, *y*, *z* e *timestamp*.

- *id* é o identificador da tabela do banco de dados, e pode ser enviado com valor 0, pois o objeto receberá o verdadeiro identificador quando for processado pelo *Gerenciados de Dados*.
- *idUsuario* é o identificador do usuário dentro da tabela de usuários no banco de dados, criada para uso futuro. O atributo *timestamp* é a data e hora da leitura, convertida para um `long` que representa o número de milissegundos desde 1 de Janeiro de 1970 até a data e hora de leitura.
- *x*, *y* e *z* são os valores decimais de uma leitura do sensor. Para acelerômetros, representam a aceleração nos eixos *x*, *y* e *z*, respectivamente. Para o Kinect, a posição da pessoa no *Skeleton Space*. Para outros sensores cuja leitura possui um número de atributos menor que 3, apenas os atributos que possuem valor devem ser preenchidos e os atributos restantes terão valor 0. Para sensores cuja leitura retorna mais de 3 atributos, deve-se armazená-los em arquivo e utilizar os métodos apropriados do arcabouço.

Código Fonte 4.2: O objeto JSON *leitura*

---

```
{"id":0,
"idUserario":1,
"x":2.9097333,
"y":6.770132,
"z":2.0355952,
"timestamp":1336134935706}
```

---

---

#### Código Fonte 4.3: Coleta e envio das leituras

---

```
int contador = 0;
final int MAX = 1000;
List<Leitura> dados = new ArrayList<Leitura>();

private void adicionaDado(Leitura dado){
    List<Leitura> copia;
    if (contador < MAX) {
        dados.add(dado);
        contador++;
    } else {
        copia = dados;
        dados.clear();
        enviar(copia);
    }
}
```

---

O envio de dados através de arquivos é mais flexível, pois é possível estender o `FileManager` para permitir a leitura de um formato de dados específico gravado em arquivo e sua posterior conversão em uma estrutura de dados a ser utilizada no arcabouço. Mais detalhes sobre a extensão do `FileManager` são apresentados na Seção 4.3.5.

### 4.3.2 Estendendo o Arcabouço

A estrutura do arcabouço permite a extensão de praticamente qualquer uma de suas partes. Esta seção mostra os passos a serem seguidos para estender as partes do arcabouço. As partes são extensíveis a partir do uso de interfaces, que reduz o acoplamento do código.

### 4.3.3 Classes de Análise de Dados

Cada classe de análise de dados estende a interface `IAnalyzer`, que agrega os métodos que definem o comportamento comum de todos os módulos analisadores de dados. Uma nova classe de análise deve implementar a interface `IAnalyzer` e prover a implementação concreta de seus métodos, definindo seu código específico de análise.

A classe abstrata `AnalyzerImpl` possui a implementação dos métodos `addReadingSet(String name, double[] data)` e `getReadingSet(String name)`, que é comum a todas as classes de análise. A nova classe de análise pode estender diretamente a classe `AnalyzerImpl` e implementar somente o método `getResults(String name)`, ou, se necessitar implementar um comportamento diferente para os dois métodos `addReadingSet` e `getReadingSet`, a classe deve implementar diretamente a interface `IAnalyzer`. O Código 4.4 mostra um exemplo de implementação de uma nova classe de análise de dados.

---

#### Código Fonte 4.4: Um Exemplo de Classe de Análise de Dados

---

```
public class GaitAnalyzer implements IAnalyzer {
    private Map<String, double[]> readings =
        new HashMap<String, double[]>();
    private DataAnalyzer dataAnalyzer;

    public AnalyzerImpl(DataAnalyzer dataAnalyzer) {
        this.dataAnalyzer = dataAnalyzer;
    }

    @Override
    public void addReadingSet(String name, double[] data) {
        readings.put(name, data);
    }

    @Override
    public double[] getReadingSet(String name) {
        return readings.get(name);
    }

    @Override
```

```
public Map<String, Double> getResults(String readingSetName) {  
    // Definir código específico de análise de dados  
}  
}
```

---

#### 4.3.4 Filtros

Novos filtros de dados podem ser adicionados, implementando a interface `IFilter`. As novas classes de filtros devem implementar a sua forma de filtragem dentro do método `execute(double[] data)`, recebendo os dados através do parâmetro `data` do método. O Código 4.5 é um exemplo da estrutura de uma classe de filtragem de dados.

---

##### Código Fonte 4.5: Um Exemplo de Classe de Filtragem de Dados

---

```
public class SimpleLowPassFilter implements IFilter {  
    @Override  
    public double[] execute(double[] data) {  
        // Realizar a filtragem  
    }  
}
```

---

#### 4.3.5 Gerenciador de Arquivos

A transferência de dados para o arcabouço através de arquivos permite uma maior liberdade para representação dos dados. A transferência de dados pelo *webservice* cobre as necessidades para a maioria dos sensores, entretanto, para formatos de dados especiais, é necessário recorrer à transferência por arquivos. A extensão do *Gerenciador de Arquivos* permite adicionar novos métodos para leitura de dados e ainda assim utilizar toda a estrutura do arcabouço. A classe `DataManager` pode ser estendida, para criar um novo *Gerenciador de Arquivos*. O Código 4.6 mostra um exemplo de como estender o *Gerenciador de Arquivos* (linha 1) e como utilizar o novo gerenciador `GaitDataFileManager` (linha 12).

---

##### Código Fonte 4.6: Um Exemplo de Extensão do Gerenciador de Arquivos

---

```
1 public class GaitDataFileManager extends FileManager {  
2
```

```
3     public List<Stride> getGaitData(String path) {
4         // Realizar a filtragem
5     }
6 }
7
8 public class UsingTheFramework {
9
10    public static void main(String[] args) {
11        DatabaseManager dbMan = new DatabaseManager();
12        FileManager fileMan = new GaitDataFileManager();
13        WriterModule writerMod = new WriterModule();
14        FilterModule filterMod = new FilterModule();
15
16        DataManager dataMan = new DataManager(dbMan, fileMan, writerMod,
17            filterMod);
18    }
```

---

### 4.3.6 Classes de Escrita de Dados

A extensão das classes de escrita de dados é feita através da implementação da interface `IWriter` ou extensão da classe `WriterImpl`, sendo o primeiro caso indicado quando se quer os métodos `writeLine`, `formatLine` e `save` com comportamentos diferentes dos métodos implementados pela classe `WriterImpl`. Um exemplo de criação de uma nova classe de escrita de dados encontra-se no Código 4.7.

---

#### Código Fonte 4.7: Um Exemplo de Classe de Escrita de Dados

---

```
public class DashSeparatedValuesWriter extends WriterImpl {

    public DashSeparatedValuesWriter(String path, String... titles)
        throws IOException {
        super(path, titles);
    }

    @Override
    public String formatLine(Object... items) {
```

---

```
        return addSeparator("-", items);  
    }  
}
```

---

## 4.4 Conclusão

Neste capítulo, foi apresentada a estrutura geral do arcabouço, detalhando suas partes e operações. Foram apresentadas as operações disponibilizadas pelo *webservice* e as classes, e seus respectivos métodos, que fazem parte do arcabouço. Por fim, foram descritas a estruturação do código na construção de um jogo e a forma de extensão do arcabouço, para adicionar novas funcionalidades.

# Capítulo 5

## Estudo de Caso

Este capítulo apresenta o desenvolvimento de dois estudos de caso para demonstrar o uso do arcabouço para captura de dados de saúde utilizando jogos eletrônicos. Os dois jogos, desenvolvidos em cooperação com o IFAL, utilizam as operações de captura de dados oferecidas pelo arcabouço. O capítulo apresenta uma descrição dos dois jogos, os tipos de dados que capturam e a forma como interagem com o arcabouço.

### 5.1 *Pinball World*

O *Pinball World* é um mini-jogo casual em terceira pessoa onde o jogador deverá controlar uma bola de *pinball* em busca de completar os objetivos do jogo. Os caminhos a serem percorridos durante o jogo apresentam obstáculos, como buracos, paredes e rampas, dos quais o jogador deve desviar a bola, coletando o máximo de estrelas de ouro que encontrar pelo caminho. O jogo se ambienta num cenário de ruínas, ao redor de plantas e uma queda d'água, transmitindo uma sensação de tranquilidade ao jogador. Na Figura 5.1, ilustram-se capturas de tela do *Pinball World*.

O jogador deverá mover a bola de *pinball* inclinando o dispositivo para esquerda, direita, frente e trás, controlando-a através dos caminhos e obstáculos, ou deixando-a descer livremente por rampas. A finalidade de controlar a bola apenas inclinando o dispositivo, através de seu acelerômetro, é capturar os níveis de tremor de movimento e repouso do jogador. O tremor de movimento é capturado durante os momentos em que o jogador precisa movimentar o dispositivo para controlar a bola e desviar de obstáculos. Em outro momento, quando o

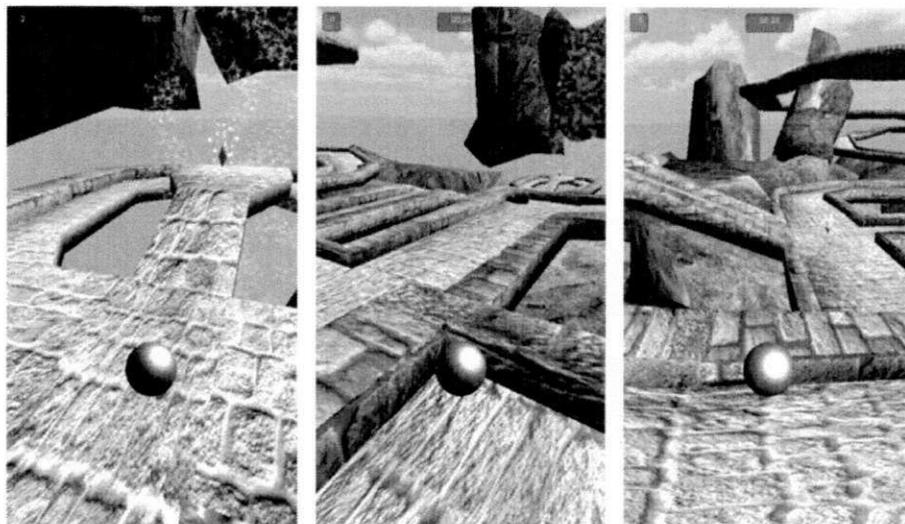


Figura 5.1: O Jogo *Pinball World*

jogador mantém o dispositivo parado e precisa apenas visualizar o trajeto da bola, é possível capturar o tremor de repouso.

### 5.1.1 Desenvolvimento do Jogo

O *Pinball World* foi desenvolvido para a plataforma Android<sup>1</sup> no *Unity*<sup>2</sup>, uma IDE (*Integrated Development Environment*) e motor de jogos 3D multi-plataforma desenvolvido pela *Unity Technologies*, que permite o desenvolvimento de jogos para *plugins* web, plataformas *desktop*, videogames e dispositivos móveis. A codificação em *Unity* é feita através de scripts, que podem ser em três linguagens, *JavaScript*, *C#* ou *Boo*. O uso de scripts dentro do *Unity* consiste de anexar objetos de script customizados chamados de *comportamentos* aos objetos do jogo. Funções específicas dentro dos objetos de script são chamadas quando ocorrem certos eventos.

As coordenadas do acelerômetro são capturadas a uma taxa de amostragem de 16Hz durante o curso do jogo, que requer conexão com a Internet para que as coordenadas sejam enviadas para o *webservice*. De acordo com a Tabela 5.1 [3], uma taxa de amostragem de 16Hz é suficiente para capturar os tipos de tremores que ocorrem nas mãos e membros. As ações de capturar as coordenadas e enviá-las para o *webservice* são imperceptíveis para

<sup>1</sup>Mais informações em: <http://www.android.com>

<sup>2</sup>Disponível em: <http://www.unity3d.com>

o jogador. No jogo as coordenadas são capturadas no método `Update`, que é o método que atualiza o desenho dos gráficos. O Código 5.1 mostra o método `Update`. A variável `taxaDeColeta` (linha 1) define o tempo em segundos entre cada leitura de dados (um total de 16 por segundo) e `quantidadeAColetar` (linha 3) define o número de leituras a serem feitas, antes de enviar as leituras para o *webservice*. O método `coletarLeituras` (linha 16) realiza as leituras de dados, transformando-os em formato JSON, e o método `enviarLeituras` (linha 19) envia os dados para o *webservice*. O Código 5.2 mostra o método `enviarLeituras`, que envia uma requisição POST para o *webservice*, através de sua operação `enviarDados`.

---

#### Código Fonte 5.1: O Método Update

---

```
1 var taxaDeColeta:float = 0.0625;
2 var proximaColeta = 0.0;
3 var quantidadeAColetar:int = 120;
4
5 function Update () {
6     acceleration = Input.acceleration;
7
8     if(Input.GetKeyDown(KeyCode.Escape)){
9         Application.Quit();
10    }
11
12    if(Time.time > proximaColeta){
13        totalColetado++;
14        proximaColeta = Time.time + taxaDeColeta;
15
16        coletarLeituras();
17
18        if(totalColetado >= quantidadeAColetar){
19            enviarLeituras();
20        }
21    }
22 }
```

---

#### Código Fonte 5.2: O Método enviarLeituras

---

```
function enviarLeituras () {
```

```
var jsonString : String = "{\ leitura \":[\"+ medicoesStr+"]}";
var postUrl = "http://" + Configuracoes.ip+":8080/accelerometer-
    webservice/enviarDados";
var encoding = new System.Text.UTF8Encoding();
var postHeader = new Hashtable();

postHeader.Add("Content-Type", "application/json");
postHeader.Add("Content-Length", jsonString.Length);

var request = WWW(postUrl, encoding.GetBytes(jsonString), postHeader);

limparLeituras();

yield request;

if (request.error != null) {
    Debug.Log("request error: " + request.error);
} else {
    Debug.Log("request success");
}
}
```

Na Figura 5.2, ilustram-se os dados do acelerômetro capturados durante uma sessão do jogo *Pinball World*. O sinal do eixo z foi filtrado usando um filtro passa alta com frequência de corte de 1 Hz.

## 5.2 *Catch the Spheres*

O *Catch the Spheres* é um mini-jogo em terceira pessoa no qual o jogador, através de seu personagem, deve capturar ou desviar de bolas que vêm em sua direção. Existem dois tipos de bolas: azuis e vermelhas. Inicialmente, todas as bolas são vermelhas e algumas destas mudam para a cor azul ao se aproximarem do jogador. O tempo para a bola mudar de cor pode ser menor ou maior, a depender do nível de dificuldade selecionado. Um personagem no centro do cenário replica todos os movimentos executados pelo jogador e capturados através do Kinect. Deve-se tocar as bolas azuis com os pés ou as mãos e desviar das bolas

Tabela 5.1: Tipos de tremor, diferenciados pela frequência, amplitude e início em relação a movimentos voluntários

Frequência	Tipo de tremor	Amplitude	Lado predominante	Relação com movimento voluntário
1-4 Hz	Cerebelar	Média-Alta	Membros	Postural, ação
3-5 Hz	Específico de tarefa	Baixa-Média	Mão	Escrita, segurar talheres, tocar instrumentos
4-5 Hz	Parkinsoniano	Média-Alta	Membros, mandíbula	Repouso
5-8 Hz	Essencial	Média-Alta	Membros, cabeça, voz	Postural
8-12 Hz	Fisiológico	Média	Membros	Postural
14-16 Hz	Ortostático	Baixa-Média	Pernas, tronco	Ficar de pé

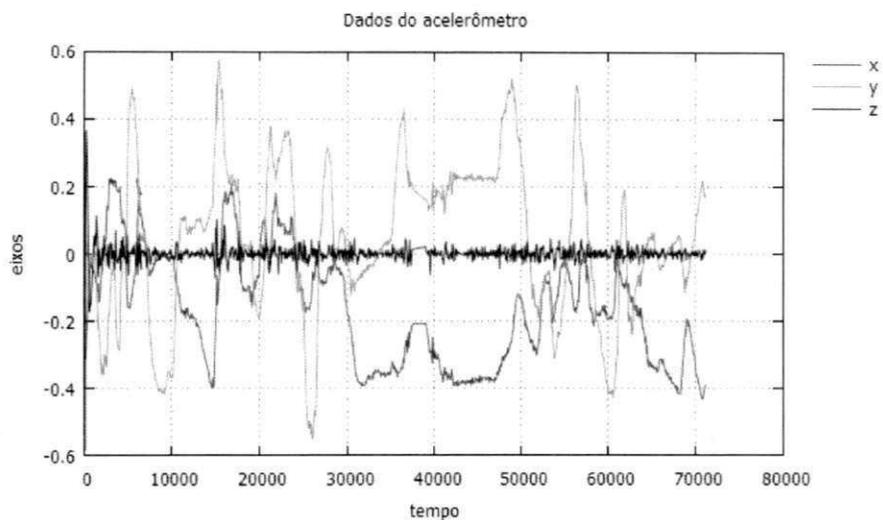


Figura 5.2: Plotagem das coordenadas da mão esquerda

vermelhas. Na Figura 5.3, ilustra-se uma captura de tela do jogo *Catch the Spheres*.

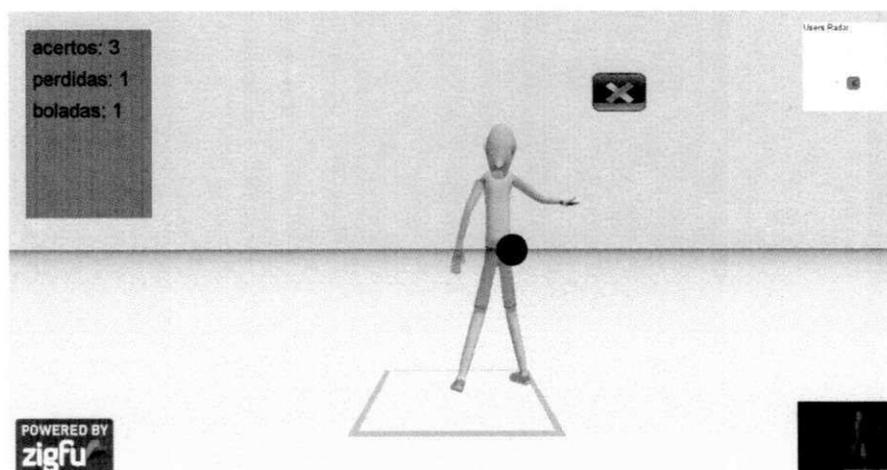


Figura 5.3: O jogo *Catch the Spheres*

A finalidade do jogo é capturar dados do movimento do jogador enquanto ele executa as ações específicas do jogo. O intervalo de tempo entre o momento em que a bola muda de cor e o momento em que a bola é capturada pelo jogador mede o reflexo do jogador, enquanto que a velocidade dos seus membros é calculada através da distância percorrida pelas mãos ou pés para capturar as bolas.

### 5.2.1 Desenvolvimento do Jogo

O *Catch the Spheres* foi desenvolvido para Windows, também no *Unity*. Uma vez que não existe suporte nativo ao Kinect no *Unity*, a integração entre o Kinect e o *Unity* é feita através do plugin *ZigFu*<sup>3</sup>.

Para *Catch the Spheres*, a linguagem de script utilizada foi JavaScript. A base de captura de dados do jogo é a classe `ZigSkeletonHealth`, cuja classe base é a `ZigSkeleton`, que define o comportamento do personagem no jogo, repetindo os movimentos do jogador de acordo com as posições das partes do corpo obtidas do Kinect. A classe `ZigSkeletonHealth` coleta as posições do jogador em tempo real através do método `UpdatePosition`, mostrado no Código 5.3. O método `UpdatePosition` é chamado sempre que uma nova posição é recebida do Kinect. A classe `ZigSkeletonHealth` intercepta a atualização das posições, primeiro gravando-as em um registro (linhas 2-6), depois enviando-as para `ZigSkeleton`, responsável por atualizar as posições do personagem dentro do jogo (linha 7).

---

#### Código Fonte 5.3: O Método `UpdatePosition`

---

```
1 public override void UpdatePosition(ZigJointId joint , Vector3 position){
2     if (logging) {
3         healthGameLog.SendMessage("NewLineHealth", SendMessageOptions.
4             DontRequireReceiver);
5         healthGameLog.SendMessage("UpdateMessageHealth", joint.ToString(),
6             SendMessageOptions.DontRequireReceiver);
7         healthGameLog.SendMessage("UpdatePositionHealth", position,
8             SendMessageOptions.DontRequireReceiver);
9     }
10    base.UpdatePosition(joint , position);
11 }
```

---

As coordenadas de posição 3D capturadas pelo Kinect são armazenadas em arquivo para serem enviadas posteriormente para o *webservice*. São coletadas as posições 3D da cabeça, pescoço, tronco, cintura, ombros, cotovelos, mãos, joelhos, quadris, tornozelo e pés. Além das posições, o jogo mantém registro dos eventos do jogo, tais como: o momento em que a bola muda de cor, o momento em que acerta o jogador, o momento em que é capturada e

---

<sup>3</sup>Disponível em: <http://zigfu.com>

qual membro tocou a bola, e o momento em que a bola é perdida pelo jogador.

Os eventos do jogo são registrados pelo *scriptEsfera.js*. Este script está acoplado ao objeto da esfera e controla todos os comportamentos relacionados a ela. No Código 5.4, é mostrado o método responsável por mudar a cor da esfera e gravar o evento de mudança de cor no registro. Esse evento de mudança de cor, junto com os outros eventos relacionados à esfera, são capturados pelo *scriptEsfera.js*.

---

#### Código Fonte 5.4: O Método mudarDeCor

---

```
function mudarDeCor(){
    renderer.material.color = Color.blue;
    healthGameLog.SendMessage("NewLineGame", SendMessageOptions.
        DontRequireReceiver);
    healthGameLog.SendMessage("NewMessageGame", "BALL-CHANGED-COLOR",
        SendMessageOptions.DontRequireReceiver);
    mudouDeCor = true;
}
```

---

O Código 5.5 mostra um trecho do registro de eventos que acontecem durante uma partida do jogo. *BALL-THROW* indica a instanciação de uma bola; *BALL INITIAL POSITION* registra a posição inicial da bola; *BALL-CHANGED-COLOR* registra o momento em que a bola muda de cor; *BALL-LOST* registra que o jogador não capturou a bola; *BALL-TOUCHED-POSITION* indica o momento em que a bola muda de cor e qual parte do corpo a tocou. No Código 5.6, é mostrado um trecho do registro de posições de cada parte do corpo durante uma partida do jogo. Cada linha mostra a data e hora, parte do corpo e as coordenadas x, y e z de sua posição. Essas informações são essenciais para que o *Módulo de Análise de Movimento* possa calcular os dados baseado nos eventos e posições do jogador.

---

#### Código Fonte 5.5: Registro de eventos do jogo

---

```
25/01/2013 14:46:02.039;BALL-THROW;
25/01/2013 14:46:02.039;BALL INITIAL POSITION;(0.7, 0.8, 15.0);
25/01/2013 14:46:04.429;BALL-LOST;
25/01/2013 14:46:06.042;BALL-THROW;
25/01/2013 14:46:06.042;BALL INITIAL POSITION;(0.7, -0.1, 15.0);
25/01/2013 14:46:08.096;BALL-CHANGED-COLOR;
25/01/2013 14:46:09.892;BALL-TOUCHED-POSITION;(-1.1, -0.3, 0.1);LeftFoot;
```

---

---

**Código Fonte 5.6: Registro de posições do jogador durante jogo**


---

25/01/2013 14:45:50.043; Head; -272.6079;882.585; -2700.646  
 25/01/2013 14:45:50.046; Neck; -271.4839;703.1649; -2718.727  
 25/01/2013 14:45:50.047; Torso; -240.7533;353.3502; -2744.491  
 25/01/2013 14:45:50.047; Waist; -234.5915;286.2368; -2706.167  
 25/01/2013 14:45:50.047; LeftShoulder; -465.3047;635.5613; -2740.658  
 25/01/2013 14:45:50.047; LeftElbow; -706.9146;748.0466; -2709.664  
 25/01/2013 14:45:50.048; LeftWrist; -974.8713;893.9808; -2650.611  
 25/01/2013 14:45:50.048; LeftHand; -1043.088;943.8502; -2600.744  
 25/01/2013 14:45:50.048; RightShoulder; -97.64243;602.8177; -2742.309  
 25/01/2013 14:45:50.049; RightElbow;27.13427;350.8448; -2744.681  
 25/01/2013 14:45:50.049; RightWrist;105.6349;94.18786; -2619.21  
 25/01/2013 14:45:50.049; RightHand;117.9995;8.172159; -2596.19  
 25/01/2013 14:45:50.050; LeftHip; -307.1635;199.0239; -2694.262  
 25/01/2013 14:45:50.050; LeftKnee; -302.8908; -343.2552; -2741.012  
 25/01/2013 14:45:50.050; LeftAnkle; -286.5235; -734.1376; -2757.292  
 25/01/2013 14:45:50.050; LeftFoot; -340.947; -798.0139; -2696.392  
 25/01/2013 14:45:50.051; RightHip; -149.0519;206.6668; -2707.094  
 25/01/2013 14:45:50.051; RightKnee; -13.03214; -357.2705; -2691.866  
 25/01/2013 14:45:50.051; RightAnkle;57.49874; -736.5664; -2653.726  
 25/01/2013 14:45:50.051; RightFoot;83.5181; -798.9075; -2605.077

---

### 5.3 Avaliação Experimental

Dezoito pessoas com idades entre 17 e 27 anos (14 homens e 4 mulheres) do Laboratório Embedded da Universidade Federal de Campina Grande e do Instituto Federal de Alagoas foram selecionadas para jogar o *Pinball World* e o *Catch the Spheres*, para testar e responder um questionário para avaliar o grau de diversão dos jogos, sua facilidade de entendimento e possibilidade de inserção de monitoramento através de jogos na rotina das pessoas. Buscou-se também validar o funcionamento do arcabouço, avaliando a possibilidade de capturar corretamente os dados de saúde. O seguinte procedimento foi utilizado para executar as sessões de teste:

1. O usuário joga o *Catch the Spheres* por, aproximadamente, 1 minuto e 30 segundos em ritmo normal.

2. Pede-se ao usuário para jogar novamente, simulando movimentos lentos, por mais 1 minuto e 30 segundos.
3. O usuário joga o *Pinball World* por, aproximadamente, 1 minuto e 30 segundos.
4. Ao final, o usuário é informado do propósito dos jogos e responde o questionário.

Para identificar a possibilidade de integrar o monitoramento da saúde do jogador através de jogos eletrônicos à sua rotina diária, foi utilizada a abordagem *Goal, Question, Metric* (GQM) [42]. GQM é uma abordagem hierárquica que inicia com objetivo principal e o divide em atividades que podem ser mensuradas durante a execução do projeto. É uma abordagem para integrar objetivos a modelos de processos de software, produtos e perspectivas de qualidade de interesse, baseado nas necessidades do projeto e da organização. Foi preparado o questionário GQM mostrado na Tabela 5.2 para avaliar a possibilidade de monitorar dados motores de forma não invasiva na rotina diária das pessoas.

Tabela 5.2: O Questionário GQM

<b>Objetivo principal:</b> Avaliar a possibilidade de monitorar dados motores de forma não invasiva na rotina diária das pessoas.
<b>Objetivo 1:</b> Avaliar o nível de diversão proporcionado pelo jogo do ponto de vista do jogador.
<b>Questão 1.1:</b> O usuário considera o jogo divertido?
<b>Métrica 1.1.1:</b> Nota em uma escala de 1 a 5 qual para quantificar o grau de diversão do jogo.
<b>Métrica 1.1.2:</b> O jogador sente-se imerso no jogo com a abordagem de captura dos movimentos? (Sim/ Não)
<b>Questão 1.2:</b> O jogador agregaria esse tipo de jogo em sua rotina diária? (Sim/ Não)
<b>Métrica 1.2.1:</b> Se o usuário tivesse adquirido esse jogo, com que frequência o utilizaria durante a semana? (1 vez/3 vezes/Todos os dias/Nunca usaria)
<b>Questão 1.3:</b> O jogo é casual?
<b>Métrica 1.3.1:</b> O jogador considera o jogo simples, sem muitas regras, de fácil entendimento e para diferentes idades? (Sim/ Não)
<b>Métrica 1.3.2:</b> O jogador se sentiria motivado a jogar esse jogo? (Sim/Não)

*Continua na próxima página*

Tabela 5.2 – Continuação da página anterior

<i>Questão 1.4:</i> O usuário costuma jogar jogos desse estilo normalmente?
<i>Métrica 1.4.1:</i> O jogador tem o costume de jogar esses jogos casuais em casa, durante uma viagem? (Sim/ Não)
<i>Questão 1.5:</i> O usuário acha que o monitoramento de saúde através de um jogo é invasivo?
<i>Métrica 1.4.1:</i> Sendo os dados acessíveis somente sob sob permissão, o usuário acha que o monitoramento de saúde através de jogos é invasivo? (Sim/ Não)
<b>Objetivo 2:</b> Avaliar os requisitos não-funcionais do jogo do ponto de vista do jogador.
<i>Questão 2.1:</i> Qual a opinião do jogador sobre o jogo?
<i>Métrica 2.1.1:</i> Nota de 1 a 5 para avaliar usabilidade.
<i>Questão 2.2:</i> Qual a opinião do jogador sobre a jogabilidade do jogo?
<i>Métrica 2.2.1:</i> Nota de 1 a 5 para avaliar jogabilidade.
<i>Questão 2.3:</i> Esse jogo seria adequado para uma criança/adulto/idoso?
<i>Métrica 2.3.1:</i> Uma criança/adulto/idoso estaria segura jogando esse jogo, ao efetuar os movimentos dos braços e das pernas? (Sim/Não)
<i>Métrica 2.3.2:</i> Qual a sua opinião do jogador sobre a faixa etária do jogo? (Livre/Crianças/Adultos/Idosos)

Tendo em vista o objetivo principal definido no questionário GQM acima, o questionário a seguir foi aplicado a 18 pessoas, na faixa de 19 a 27 anos, perguntando-as sobre sua opinião em relação ao jogo. A preocupação principal foi avaliar o grau de entretenimento dos jogadores, a possibilidade de integrar jogos para monitoramento na rotina dos jogadores, motivação para jogar, segurança e opinião do jogador em relação ao monitoramento da saúde.

1. Numa escala de 1 a 5 qual o grau de diversão do jogo?
2. Você agregaria um jogo desse estilo em sua rotina diária? (Sim/Não)
3. Se você tivesse adquirido esse jogo, com que frequência você o utilizaria durante a semana? (1 vez/3 vezes/Todo dia/Nunca)

4. Você considera o jogo simples, sem muitas regras, de fácil entedimento e para diferentes idades? (Sim/Não)
5. Você sentiria motivado a jogar esse jogo? (Sim/Não)
6. Você tem o costume de jogar esses jogos casuais? (Sim/Não)
7. Uma criança estaria segura jogando esse jogo, ao efetuar os movimentos dos braços e das pernas (Sim/Não)
8. Um adulto estaria seguro ao jogar esse jogo, ao efetuar os movimentos dos braços e das pernas (Sim/Não)
9. Um idoso estaria seguro ao jogar esse jogo, ao efetuar os movimentos dos braços e das pernas (Sim/Não)
10. A qual faixa etária você acha o jogo é direcionado? (Livre/Crianças/Adultos/Idosos)
11. Sendo os dados acessíveis somente sob sob permissão, você acha que o monitoramento de saúde através de jogos é invasivo? (Sim/Não)

Os resultados do questionário são apresentados na Figura 5.4. O resultado foi positivo, considerando as respostas relativas à aceitação dos jogos pelas pessoas. Com as respostas positivas obtidas foram atingidos os dois objetivos definidos no questionário GQM:

**Objetivo 1** Avaliar o nível de diversão proporcionado pelo jogo do ponto de vista do jogador.

**Objetivo 2** Avaliar os requisitos não-funcionais do jogo do ponto de vista do jogador.

O Objetivo 1 pode ser considerado alcançado, devido às respostas alcançadas com as perguntas 1, 2, 3 e 5 do questionário. Das 18 pessoas, 61% (11 pessoas) acharam os jogos divertidos, atribuindo a eles uma nota 4 (de 1 a 5), 67% (12 pessoas) agregariam jogos casuais à sua rotina diária, 61% (11 pessoas) os jogariam 3 vezes por semana e 94% (17 pessoas) sentir-se-iam motivadas a jogar os jogos. Através dessas respostas, é possível concluir que jogos casuais no estilo dos apresentados nesse estudo poderiam proporcionar diversão aos jogadores, enquanto realizam o monitoramento em sua rotina diária.

Tabela 5.3: Variação de Velocidade em Duas Sessões do Jogo *Catch the Spheres*

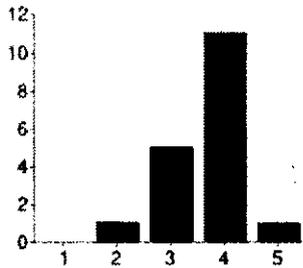
VM (Movimento rápido)	DP	VM (Movimento lento)	DP	Variação Média	DP
35,32cm/s	6,34	21,83cm/s	7,42	13,49cm/s	1,6

Quanto ao Objetivo 2, a resposta dos jogadores indica que os jogos possuem características como facilidade de entendimento e segurança, apesar de requererem que o usuário jogue movimentando-se. Todos os jogadores consideraram o jogo de fácil entendimento e que é seguro para crianças e adultos. Entretanto, 72% (13 pessoas) acha que os jogos não seriam seguros para idosos. Um total de 80% acha que monitorar a saúde através de jogos não é um procedimento invasivo. A aceitação dos usuários e opinião quanto à facilidade de uso tornam possível concluir que o monitoramento através de jogos poderia ser realizado sem maiores problemas, capturando corretamente os dados de saúde dos jogadores.

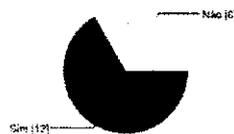
Apesar de o questionário ter avaliado a opinião dos jogadores quanto aos dois jogos apresentados, é possível generalizar e dizer que essas opiniões são válidas para outros jogos no mesmo estilo, desde que possuam um enredo atrativo. Um projeto de jogo melhor elaborado poderia ter obtido uma aceitação ainda mais positiva.

Ao analisar o movimento normal e o movimento lento de todos os usuários durante o jogo *Catch the Spheres*, foi possível constatar a redução ou, no mínimo, uma alteração na velocidade de movimento. Após remover as amostras de dados inválidas (por exemplo, aquelas em que o jogador já estava com o membro na posição exata de capturar a bola), foi calculada a média (VM) e desvio padrão (DP) da velocidade dos membros dos jogadores, mostrados na Tabela 5.3.

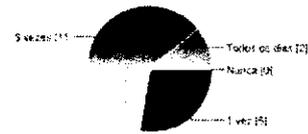
Também, com a análise dos dados coletados durante as sessões do *Pinball World*, foi possível verificar que nenhum dos usuários possui amplitude de tremor significativa. A média das frequências de tremor é de 2,85Hz, com magnitude média de 3,15. Os desvios padrão para a magnitude e frequência, respectivamente, são: 1,24 e 1,7, indicando que a média da magnitude e frequência de tremor dos jogadores é bastante próxima.



(a) Pergunta 1 (1: 0%; 2: 6%; 3: 28%; 4: 61%; 5: 6%)



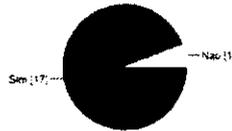
(b) Pergunta 2 (Sim: 67%; Não: 33%)



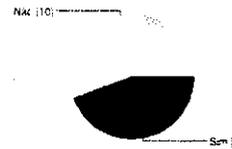
(c) Pergunta 3 (1 vez: 28%; 3 vezes: 61%; Todos os dias: 11%; Nunca: 0%)



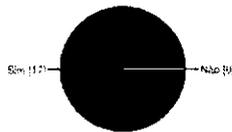
(d) Pergunta 4 (Sim: 100%)



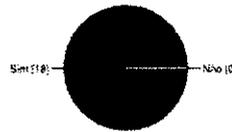
(e) Pergunta 5 (Sim: 94%; Não: 6%)



(f) Pergunta 6 (Sim: 44%; Não: 56%)



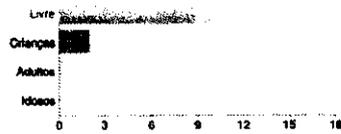
(g) Pergunta 7 (Sim: 100%)



(h) Pergunta 8 (Sim: 100%)



(i) Pergunta 9 (Sim: 72%; Não: 28%)



(j) Pergunta 10 (Livres: 80%;

Crianças: 10%; Adultos: 10%; Idosos: 0%)



(k) Pergunta 11 (Sim: 17%; Não: 83%)

Figura 5.4: Respostas do questionário

## **5.4 Conclusão**

Este capítulo mostrou como exemplos de jogos simples podem explorar superficialmente o potencial do arcabouço. Jogos mais complexos podem contemplar ações para prover um suporte mais completo ao monitoramento da saúde. Um projeto de jogo mais elaborado, aliado a uma melhor apresentação gráfica, entre outros itens de projetos de jogos que mantêm a atenção do jogador, podem criar uma experiência de monitoramento contínua e prolongada.

## Capítulo 6

### Considerações Finais

Ultimamente, têm surgido diversos jogos com o propósito de motivar o jogador a exercitar-se ou realizar monitoramento. Entretanto, estas soluções requerem que o jogador habitue-se com um estilo de jogo que, geralmente, não está presente em seu cotidiano. Além disso, as ações realizadas dentro do jogo são incomuns para o tipo de jogo geralmente preferido pelos jogadores.

A contribuição desse trabalho é um arcabouço que permite realizar a aquisição de dados de saúde em jogos eletrônicos. Sua forma de captura permite que ele receba os dados de monitoramento de qualquer jogo, se os dados estiverem no formato requerido pelo arcabouço, em JSON, ou um formato específico definido pelo desenvolvedor. É necessário apenas adaptar os movimentos necessários do jogador durante o jogo, de forma que os sensores de movimento capturem os dados corretamente para obtenção de informação de saúde. A inclusão da aquisição de dados de saúde de forma invisível para o usuário permite realizar o monitoramento em jogos não direcionados para a saúde, como os de ação, aventura, *arcade*, etc.

De acordo com as respostas obtidas dos jogadores durante os testes, os jogadores consideraram os jogos desenvolvidos divertidos e de fácil entendimento. A maioria deles declarou sentir-se motivada a jogar o jogo, incluindo-o em sua rotina diária. Apesar dos experimentos não terem sido realizados com pessoas doentes, devido à dificuldade em se conseguir autorização do Conselho de Ética, através dos experimentos realizados foi possível verificar que, se ocorrerem movimentos incomuns, é possível detectá-los através dos métodos corretos.

Alguns sensores ainda não suportados pelo arcabouço, como monitor de batimentos car-

---

díacos vestível e sensor de peso, já são utilizados no contexto de jogos e isso abre margem para extensão e evolução do arcabouço, gerando novas soluções de monitoramento no contexto de assistência médica pervasiva. A partir do uso do arcabouço, torna-se simples o suporte ao uso desses sensores na rotina do indivíduo de forma não perturbadora, sem afetar a sua rotina diária.

Através de técnicas de aprendizado de máquina, como redes neurais e máquinas de vetores de suporte, é possível classificar os dados e obter informações que não são descobertas somente com o uso de regras. A adição de um módulo de aprendizado de máquina é uma extensão valiosa do arcabouço.

Outro trabalho futuro a ser realizado é adicionar um módulo estatístico ao arcabouço, que encontre medidas estatísticas para os parâmetros capturados, informando qual o intervalo de normalidade para determinados valores. O módulo estatístico pode prover funções úteis, como verificar se os movimentos do usuário estão dentro do normal, quando comparados a uma amostra, dentro de uma população.

## Bibliografia

- [1] Rikke Aarhus and Stinne Aaløkke Ballegaard. Negotiating boundaries. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, pages 1223–1232, New York, New York, USA, 2010. ACM Press.
- [2] Sheikh I Ahamed, Nilothpal Talukder, and Achilles D Kameas. Towards privacy protection in pervasive healthcare. In *Intelligent Environments, 2007. IE 07. 3rd IET International Conference on*, pages 296–303. IET, 2007.
- [3] Alberto Albanese and Joseph Jankovic. Distinguishing clinical features of hyperkinetic disorders. *Hyperkinetic Movement Disorders: Differential Diagnosis and Treatment*, page 1, 2012.
- [4] Hande Alemdar and Cem Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688–2710, October 2010.
- [5] Leonard A Annetta. The i’sh’ve it: A framework for serious educational game design. *Review of General Psychology*, 14(2):105, 2010.
- [6] Aurelie Arntzen. Game based learning to enhance cognitive and physical capabilities of elderly people: Concepts and requirements. *World Academy of Science Engineering and Technology*, 2011.
- [7] Oliver Assad, Robert Hermann, Damian Lilla, Björn Mellies, Ronald Meyer, Liron Shevach, Sandra Siegel, Melanie Springer, Saranat Tiemkeo, Jens Voges, and Others. WuppDi! Supporting Physiotherapy of Parkinsons Disease Patients via Motion-based Gaming. In *Mensch and Computer 2011: 11. Fachübergreifende Konferenz für Interaktive und Kooperative Medien über Medienjubermorgen*, page 469, 2011.

- [8] Louis Atallah, Benny Lo, and Guang-Zhong Yang. Can pervasive sensing address current challenges in global healthcare? *Journal of Epidemiology and Global Health*, 2(1):1–13, March 2012.
- [9] Stephen D Atkinson and V Lakshmi Narasimhan. Design of an introductory medical gaming environment for diagnosis and management of Parkinson’s disease. In *Trendz in Information Sciences & Computing (TISC), 2010*, pages 94–102. IEEE, 2010.
- [10] Juan Carlos Augusto, Norman D. Black, H. Gerry McAllister, Paul J. McCullagh, and Christopher D. Nugent. Pervasive health management: New challenges for health informatics. *J. UCS*, 12(1):1–5, jan 2006.
- [11] Chris R Baker, Kenneth Armijo, Simon Belka, Merwan Benhabib, Vikas Bhargava, Nathan Burkhart, Artin Der Minassians, Gunes Dervisoglu, Lilia Gutnik, M Brent Haick, et al. Wireless sensor networks for home health care. In *Advanced Information Networking and Applications Workshops, 2007, AINAW’07. 21st International Conference on*, volume 2, pages 832–837. IEEE, 2007.
- [12] Paolo Bonato, Delsey M Sherrill, David G Standaert, Sara S Salles, and Metin Akay. Data mining techniques to detect motor fluctuations in parkinson’s disease. In *Engineering in Medicine and Biology Society, 2004. IEMBS’04. 26th Annual International Conference of the IEEE*, volume 2, pages 4766–4769. IEEE, 2004.
- [13] Ross A. Clark, Yong-Hao Pua, Karine Fortin, Callan Ritchie, Kate E. Webster, Linda Denehy, and Adam L. Bryant. Validity of the microsoft kinect for assessment of postural control. *Gait & Posture*, 36(3):372 – 377, 2012.
- [14] Z.S. de Urturi, A.M. Zorrilla, and B.G. Zapirain. Serious game based on first aid education for individuals with autism spectrum disorder (asd) using android mobile devices. In *Computer Games (CGAMES), 2011 16th International Conference on*, pages 223 –227, july 2011.
- [15] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: defining "gamification". In *Proceedings of the 15th Inter-*

- national Academic MindTrek Conference: Envisioning Future Media Environments*, MindTrek '11, pages 9–15, New York, NY, USA, 2011. ACM.
- [16] Rodger J Elble. Gravitational artifact in accelerometric measurements of tremor. *Clinical Neurophysiology*, 116(7):1638 – 1643, 2005.
- [17] Moshe Gabel, Erin Renshaw, Assaf Schuster, and Ran Gilad-Bachrach. Full body gait analysis with kinect. In *Proceedings of the IEEE Engineering in Medicine and Biology Society 2012*. IEEE, 2012.
- [18] Stefan Göbel, Sandro Hardy, Viktor Wendel, Florian Mehm, and Ralf Steinmetz. Serious games for health: personalized exergames. In *Proceedings of the international conference on Multimedia*, pages 1663–1666. ACM, 2010.
- [19] F Gouaux, L Simon-Chautemps, S Adami, M Arzi, D Assanelli, J Fayn, MC Forlini, C Malossi, A Martinez, J Placide, et al. Smart devices for the early detection and interpretation of cardiological syndromes. In *Information Technology Applications in Biomedicine, 2003. 4th International IEEE EMBS Special Topic Conference on*, pages 291–294. IEEE, 2003.
- [20] Sandro Hardy, Abdulmotaieb El Saddik, Stefan Gobel, and Ralf Steinmetz. Context aware serious games framework for sport and health. In *2011 IEEE International Symposium on Medical Measurements and Applications*, pages 248–252. IEEE, May 2011.
- [21] Holly Jimison, Misha Pavel, and Thai Le. Home-based cognitive monitoring using embedded measures of verbal fluency in a computer word game. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 3312–3315. IEEE, 2008.
- [22] Pamela M. Kato. Video games in health care: Closing the gap. *Review of General Psychology*, 14(2):113–121, 2010.
- [23] Gabor Kiss. Using smartphones in healthcare and to save lives. In *Internet of Things (iThings/CPSCoM), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, pages 614–619. IEEE, 2011.