

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CARLOS TORTURELLA VALADÃO

**USO DA ROBÓTICA PARA EXPLORAÇÃO E DESCOBERTA DO
AMBIENTE PARA CRIANÇAS COM DEFICIÊNCIA FÍSICA
SEVERA**

VITÓRIA

2011

CARLOS TORTURELLA VALADÃO

**USO DA ROBÓTICA PARA EXPLORAÇÃO E DESCOBERTA DO
AMBIENTE PARA CRIANÇAS COM DEFICIÊNCIA FÍSICA
SEVERA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito para obtenção do Grau de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Teodiano Freire Bastos Filho.

VITÓRIA
2011

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

V136u Valadão, Carlos Torturella, 1986-
 Uso da robótica para exploração e descoberta do ambiente
 para crianças com deficiência física severa / Carlos Torturella
 Valadão. – 2011.
 160 f. : il.

Orientador: Teodiano Freire Bastos Filho.
Dissertação (Mestrado em Engenharia Elétrica) – Universidade
Federal do Espírito Santo, Centro Tecnológico.

1. Robótica. 2. Crianças deficientes. 3. Desenvolvimento
cognitivo. 4. Auto-estima em crianças. I. Bastos Filho, Teodiano
Freire. II. Universidade Federal do Espírito Santo. Centro
Tecnológico. III. Título.

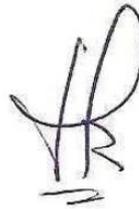
CDU: 621.3

CARLOS TORTURELLA VALADÃO

**USO DA ROBÓTICA PARA EXPLORAÇÃO E DESCOBERTA DO AMBIENTE
PARA CRIANÇAS COM DEFICIÊNCIA FÍSICA SEVERA**

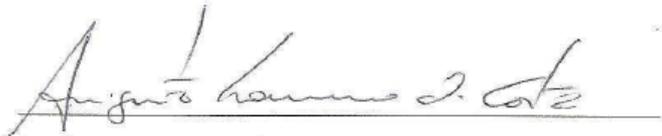
Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito para a obtenção do Grau de Mestre em Engenharia Elétrica.

COMISSÃO EXAMINADORA:



Prof. Dr. Teodiano Freire Bastos Filho - Orientador

Universidade Federal do Espírito Santo



Prof. Dr. Augusto César Loureiro

Universidade Federal da Bahia



Prof.^a Dr.^a Agda Gonçalves

Universidade Federal do Espírito Santo

*A toda a minha família, amigos e a todas as
crianças que possam se beneficiar com este projeto.*

Agradecimentos

Agradecer é de fundamental importância. É um reconhecimento e gratidão pelas pessoas amigas e acolhedoras que ajudaram.

Por isso, agradeço a Deus que, com toda certeza, me iluminou, assim como todos os participantes deste projeto que visa beneficiar crianças com dificuldades motoras.

Agradeço também à minha família que me deu um grande apoio, tanto nas horas alegres como nos momentos complicados. Ao meu orientador pelo seu esforço para tornar este projeto melhor e por me ajudar a buscar cada vez mais conhecimento.

Também sou muito grato aos meus amigos Victor Perim, Magdo Bôrtole, Fauzi Rodor, Daniel Celino e Jhon Freddy Sarmiento que me auxiliaram de forma substancial na realização deste projeto. Aos colegas do laboratório e aos professores André Ferreira e Anselmo Frizzera também sou grato pelo apoio.

Agradeço também pelo auxílio dado pelo Prof. Humberto Ferasoli, da Unesp de Bauru-SP, à Prof. Natalia Lopez da Universidade Nacional de San Juan-Argentina e também à Pedagoga Prof.^a Agda Gonçalves, do CEUNES/UFES de São Mateus.

Por fim, os meus agradecimentos à FACITEC/PMV pelo financiamento concedido para a realização deste projeto de pesquisa e pela bolsa de mestrado concedida, aos pais das crianças que foram voluntárias, pela confiança no projeto e aos profissionais de saúde do Comitê de Ética e do Centro de Reabilitação e Fisioterapia do Espírito Santo (CREFES) por permitirem a realização mesmo.

A todos vocês, muito obrigado!

Resumo

Crianças com deficiência física severa têm pouca chance de conseguirem interagir com o mundo e, devido a essa falta de interação, acabam por criar um conceito de que não são capazes de realizar tarefas sozinhas, idéia esta chamada de “indefesa aprendida”. Com esta falta de experiências, pode ocorrer comprometimento na parte de aprendizagem, uma vez que uma boa interação com o ambiente ao redor auxilia muito no processo de aprendizado. No entanto, com o auxílio de robôs, é possível fazer com que haja uma interação com o mundo de uma forma diferente, melhorando sua capacidade de cognição, socialização e, por fim, sua própria auto-estima. Tais elementos são fundamentais para uma melhor qualidade de vida.

A grande motivação desta Dissertação de Mestrado é fazer com que essas crianças tenham uma melhor qualidade de vida, e, principalmente, desenvolvimento cognitivo e auto-estima. O sistema desenvolvido permite que a criança manipule de forma mais autônoma robôs, desde que configurados previamente pelo programa. Para fazer tal manipulação a criança utiliza alguns dispositivos auxiliares como sensores de inclinação e sensores mioelétricos, dentre outros dispositivos que podem ser configurados no sistema.

Isso faz com que o sistema seja flexível a diversos dispositivos e robôs, bastando apenas fazer as devidas configurações, tanto no dispositivo como no sistema. É esperado que com tal programa as crianças sejam capazes de ter maior interação social, melhoria no quadro cognitivo e auto-estima, tentando, de certa forma, equilibrar os efeitos negativos da deficiência física.

Abstract

Severe-physically disabled children have little chance of environmental and social exploration and discovery, and due this lack of interaction and independency, this fact may lead to an idea that they are unable to do anything by themselves. This idea is called learned helplessness and is very negative for the child's cognitive development and social development as well. With this entire situation it is very likely that the self-esteem and mood will not be good and also there is a high probability of psychological problems.

Trying to help children in this situation, educational robotics can offer and aid, once it can give them a certain degree of independency in exploration of environment. The system allows the child to control a sensor device that transmits the collected data to a computer which processes and sends the command to the robot. The data collected can be from head movement, muscle signals or any other body signal supported by the device. The configuration of the device must be done inside the software developed for this project. There is an expectation that using this system the negative effects provoked by disabilities may be softened, in a certain degree, and the child can have a better life and specially a better self image and self-esteem.

Results are shown in the last chapter and the collected data about the volunteer children are able to show how important the usage of this system is on the performed tests for their development.

Sumário

Lista de Figuras	11
Capítulo 1: Introdução.....	12
1.1 Learned Helplessness	13
1.2 Aspectos importantes a considerar	19
Capítulo 2: Métodos para avaliação	21
2.1 <i>Goal Attainment Scale</i> (GAS).....	21
2.2 Medição com base no comportamento.....	22
Capítulo 3: Hardware	23
3.1 Robô Móvel com Manipulador	23
3.1.1 POB-PROTO – A placa de controle.....	26
3.1.2 Placa de recepção Bluetooth.....	27
3.1.3 Alterações realizadas no robô.....	28
3.2 Placa de aquisição de sinais	29
3.3 Câmera Externa.....	30
Capítulo 4: Funcionamento do Sistema	32
4.1 Parte 1: Aquisição e conversão dos sinais.....	32
4.2 Parte 2: Transmissão de dados via Bluetooth.....	34
4.3 Parte 3: Interface	35
4.3.1 Interface – Funcionamento geral.....	35
4.4 Parte 4: Interface – Organização dos dados	38
4.5 Cadastros e conversão de dados.....	43
4.6 Tratamento do sinal de sEMG.....	44
4.7 Calibração do sensor de inclinação	45
4.8 Modos possíveis de operação	47
4.9 Relatórios automáticos	48
Capítulo 5: Tarefas.....	51
□ Tarefa inicial (treino) – Mover o robô através de caminhos com obstáculos.....	51
□ Tarefa #1 – Manipulação do robô para que façam um desenho.....	51
□ Tarefa #2 – Trajetórias com blocos coloridos	52
6.1 Utilização do método de avaliação GAS.....	54
7.1 Resultados observados	57
7.2 Resultados estatísticos por semana	62
7.3 Comparações entre as semanas	68
Capítulo 8: Conclusões	73

Referências Bibliográficas	75
Apêndice I: Programação feita em Visual Basic.....	78
<input type="checkbox"/> Módulo Global #1 – Variáveis Globais.....	78
<input type="checkbox"/> Módulo Global #2 – Funções Globais.....	83
<input type="checkbox"/> Tela inicial – O Menu.....	87
<input type="checkbox"/> Janela principal – Tarefas.....	90
<input type="checkbox"/> Cadastro de crianças.....	109
<input type="checkbox"/> Cadastro de tarefas	113
<input type="checkbox"/> Cadastro de robôs	118
<input type="checkbox"/> Cadastro de dispositivos auxiliares	121
<input type="checkbox"/> Relatórios	125
<input type="checkbox"/> FastCalibration	135
<input type="checkbox"/> Sequência	137
<input type="checkbox"/> SplashScreen	141
<input type="checkbox"/> AboutBox	142
Apêndice II: Programação feita no PIC (POB-BOT).....	144
Apêndice III: Recursos extras do robô.....	146
<input type="checkbox"/> POB-LCD – Display LCD	146
<input type="checkbox"/> POB-EYE – Placa da câmera do robô.....	146
Apêndice IV: Carta de aprovação do Comitê de Ética.....	148
Apêndice V: Carta para o Comitê de Ética da Universidade Federal do Espírito Santo.....	149
Apêndice VI: Formulário de Consentimento dos Responsáveis pelos Voluntários.....	151
Apêndice VII: Convite para Participação do Projeto de Pesquisa	153
Apêndice VIII: Documentação para o Conselho Nacional de Saúde e CONEP	158

Lista de Figuras

Figura 1 - Processo de “indefesa aprendida” (<i>learned helplessness</i>).	14
Figura 2 - Auxílio da robótica para impedir a idéia de “indefesa aprendida”.	14
Figura 3 - Foto do robô POB-EYE com Bluetooth conectado (parte lateral).	23
Figura 4 - Robô no plano xy	24
Figura 5 - Placa POB-PROTO [16].	26
Figura 6 - Placa de recepção Bluetooth.	27
Figura 7 - Diagrama esquemático da placa de recepção [17].	28
Figura 8 – Componentes extras.	29
Figura 9 – Sensores	30
Figura 10 - Webcam	31
Figura 11 - Esquema utilizado para captura dos sinais e movimentação do robô.....	32
Figura 12 - Sensor junto com os suportes utilizados nos testes.....	33
Figura 13 - Forma como as informações são enviadas pelo sensor.....	34
Figura 14 - Prioridade dos eixos.....	36
Figura 15 - Caracteres que representam cada movimento do robô.	37
Figura 16 - Esquema simplificado de como são gerados os relatórios de testes.	39
Figura 17 - Representação gráfica dos bancos de dados	40
Figura 18 – Representação gráfica das informações guardadas nos relatórios	40
Figura 19 - Tela principal do software de robótica educacional.	41
Figura 20 - Esquema simplificado de funcionamento do sistema.	42
Figura 21 – Tradução (conversão de dados).....	44
Figura 22 - Tela responsável pelas calibrações.	46
Figura 23 – Relatório gerado com o programa.....	50
Figura 24 – Desenho com o robô.....	52
Figura 25 - Tarefa de trajetória com blocos coloridos.....	52
Figura 26 - Média de movimentos feitas por cada criança durante a primeira tarefa. ...	63
Figura 27 - Tempo médio que cada criança demorou no treino.	64
Figura 28 - Quantidade de movimentos feitos por segundo por cada criança.....	64
Figura 29 - Média de movimentos feitos por cada criança na segunda semana.....	65
Figura 30 - Tempo médio para execução da tarefa (por criança) na segunda semana. ..	66
Figura 31 - Número de movimentos médio por segundo na segunda semana.	66
Figura 32 - Média de movimentações feita na última semana.	67
Figura 33 - Tempo médio dos testes na última semana.....	68
Figura 34 - Quantidade de movimentos por segundo por criança na última semana.	68
Figura 35 - Tempo médio dos testes na última semana.....	69
Figura 36 - Tempo médio dos testes na última semana.....	71
Figura 37 - Média de movimentos válidos por criança nas três semanas.....	72
Figura 38 - Placa POB-LCD.....	146
Figura 39 - Placa POB-EYE.....	147
Figura 40 - Carta de aprovação do comitê de ética.	148

Capítulo 1: Introdução

Na infância a criança aprende pelas descobertas, que são feitas através da exploração do mundo em que vive. Os dois primeiros anos da vida de um indivíduo são considerados como período senso-motor [1,2]. Durante esse estágio de desenvolvimento, a interação com o ambiente é feita através de sensações físicas e estímulos nos sentidos [1,3]. Isto significa que aprendem a se relacionar com seus próprios corpos e com o ambiente, sendo que tal aprendizado se dá por meio de experiências repetitivas e por exploração do mundo através de seus sentidos [1]. Ao término do período senso-motor, as crianças têm uma idéia de “permanência do objeto”, posição de “objetos” no espaço e tempo, além de algumas relações entre eles [1]. “Permanência do objeto” é entendido como a compreensão da existência de objetos no espaço e o fato do mesmo permanecer nesse espaço.

Vários pesquisadores do presente e do passado estudam o comportamento das crianças, dentre os quais pode-se citar Ley Vigotski, que foi um psicólogo bielo-russo pioneiro na descoberta da interação das crianças com o meio e seu desenvolvimento intelectual [4].

Sendo essencial a manipulação de objetos e interação com o meio ambiente, as crianças com deficiência física severa são privadas de maiores oportunidades de explorar o mundo como as outras crianças, ficando assim muito aquém do que o ambiente pode oferecer em termos de aprendizado através da exploração [1].

Com a falta de independência, exploração e espontaneidade na descoberta dos elementos que as cercam, tais crianças podem ter influência tanto na questão da aprendizagem, como no desenvolvimento social da mesma. Toda essa situação de falta de estímulos acaba por provocar uma infância tardia [1].

Tendo limitações, as crianças com deficiência são, normalmente, dependentes de seus pais ou cuidadores para interagirem com o mundo. Brinker e Lewis, [1,5], sugerem que o comportamento de crianças com deficiência pode, por si só, definir as escolhas dos pais e cuidadores sobre quais situações no ambiente a criança irá explorar. Essas escolhas acabam restringindo e comprometendo o desenvolvimento cognitivo e social da criança.

1.1 Learned Helplessness

Todo esse cenário acaba por provocar uma perda de interesse na exploração do ambiente e pode desenvolver a idéia de “indefesa aprendida” ou *learned helplessness* [1], em inglês, que significa que a criança se acha incapaz de conseguir fazer algo sem qualquer auxílio externo e de ter controle sobre suas ações. Com a “indefesa aprendida”, a criança normalmente adota uma passividade e desinteresse em relação ao meio [1,6]. A idéia oposta da indefesa aprendida é a idéia de *mastery motivation*, o que significa uma grande motivação, na qual a criança se sente capaz de realizar as ações que pretende por si só [1,6].

De acordo com Seligman, [1,7], a indefesa aprendida faz com que a própria criança tenda a evitar situações que, em sua mente, apresentem algum grau de risco, dificuldade, ou seja, algo que provoque desgosto. Também é sugerido por [7] que tal comportamento acaba levando à depressão clínica e a outros problemas psicológicos devido à percepção da falta de controle que se tem do ambiente.

Todos esses fatores acabam por comprometer o comportamento das crianças, tornando-as passivas, dependentes e socialmente inativas. Para evitar que isso ocorra é necessário que suas necessidades de desenvolvimento específicas sejam reconhecidas e que sejam dadas a essas crianças oportunidade de explorar o mundo ao seu redor, através de métodos alternativos [1,8]. Com essa atitude é possível uma motivação maior, dando à criança a oportunidade de experimentar o seu comportamento e de receber estímulos e, assim, diminuir a idéia de “indefesa aprendida” [1,6].

A figura 1 mostra a representação de como ocorre o processo da indefesa aprendida.



Figura 1 - Processo de “indefesa aprendida” (*learned helplessness*).

Algumas tecnologias de assistência às crianças com deficiências motoras severas têm possibilitado que elas sejam capazes de ter um controle sobre o ambiente [1]. No diagrama exibido na figura 2 é explicado como a robótica pode ajudar a reduzir, ou até mesmo acabar, com a idéia da indefesa aprendida.

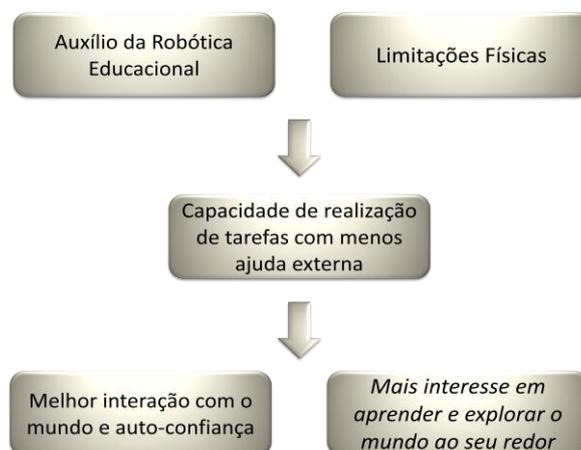


Figura 2 - Auxílio da robótica para impedir a idéia de “indefesa aprendida”.

O trabalho de Swinth, Anson & Deitz [1,9] mostra que crianças a partir dos seis anos de idade já têm condições de acessar um programa de causa e efeito, controlado por computador, através de acionamento de um interruptor. Quanto mais cedo essas crianças forem expostas a este tipo de tecnologia, mais fraca será a idéia da indefesa aprendida e tal exposição deve ser feita durante e após o período considerado crítico

(período sensório motor – até aproximadamente 2 anos, como descrito por Piaget) [1,10].

Com a utilização da Robótica e Tecnologias Assistivas (tecnologias que ajudam pessoas portadoras de deficiência física ou doenças) é possível prover às crianças com deficiências físicas uma grande oportunidade de escolher como elas querem interagir com o meio e, ainda, a possibilidade de exercer certo controle sobre o mesmo, ao invés de terem uma atividade forçada designada para elas [5]. Além disso, sistemas robóticos também permitem a manipulação de objetos reais (em três dimensões), ao invés de gráficos computacionais (duas dimensões) para manipulação simulada, fazendo-as interagir de forma mais real [1].

Vários estudos foram encontrados na literatura para determinar se crianças muito jovens poderiam interagir com um manipulador robótico [11]. Nove crianças, sendo seis com deficiência e três sem deficiência, todas com menos de 38 meses de vida participaram dessa pesquisa (contida na literatura em [11]). O sistema consistia em um computador para controlar e coletar dados e um pequeno manipulador robótico [11]. O manipulador foi utilizado pelas crianças como se fosse uma ferramenta, de forma que pressionando um interruptor poderiam trazer um objeto para mais próximo de si ou descobrir um objeto que estivesse escondido [1,11,12].

Cinquenta por cento das crianças com deficiência (todas com desenvolvimento cognitivo de sete a nove meses ou mais) e cem por cento das crianças sem deficiência interagiram com o manipulador robótico, usando-o como uma ferramenta para pegar objetos fora do alcance delas. Foi observado que o nível cognitivo e lingüístico das crianças era muito maior do que o nível motor [1,11].

Posteriormente, esse trabalho prévio foi estendido, sempre com o foco na exploração e descobrimento do ambiente pelas crianças. Seus objetivos eram [1]:

- Avaliar como as crianças com deficiências físicas severas usam um manipulador robótico para exploração;
- Determinar qual a relação existente entre as escolhas feitas através do interruptor e a complexidade dos resultados das tarefas realizadas.

Esse estudo de capacidade mostrou de qual forma as crianças (com idade entre três a seis anos) com deficiência utilizam o manipulador robótico para exploração do ambiente. O mesmo foi usado em um formato exploratório, no qual as crianças tinham acesso a vários movimentos através do ato de pressionar um ou mais interruptores [13].

Uma série progressiva de tarefas mais complexas fez com que houvesse um aumento do nível do desenvolvimento das mesmas. Viu-se, então, a necessidade de mais interruptores (que estavam pré-programados) para a criança conseguir completar a tarefa [1,8,11,12]. Foi mostrado para a criança como o manipulador robótico se movimenta, através de uma única ativação do interruptor, de forma a demonstrar o movimento prévio e encorajá-las a pressionar o mesmo [1,10]. Para maximizar as experiências das crianças, foi utilizada uma vareta dentro de um tubo, com a finalidade de prover tanto interações motoras como sensoriais [1]. As tarefas utilizadas foram:

- O manipulador robótico deixa cair as varetas em um copo, pressionando o interruptor uma única vez;
- A criança controla o manipulador robótico para encher o copo com varetas, pegá-las e deixá-las cair, utilizando dois interruptores;
- A criança faz com que o manipulador robótico se movimente lateralmente, enchendo o copo com varetas, levanta o copo e deixa-o cair dentro de um tubo, usando três interruptores.

Cada sessão com a criança foi gravada em vídeo para revisão [1,5,11,10]. As observações do estudo incluíram as ações da criança e os comportamentos das mesmas durante a realização das tarefas. Por exemplo, olhando em direção ao manipulador robótico, olhando para o interruptor, verificando respostas positivas ou negativas (exemplo: rindo ou chorando) em antecipação ou em resposta à ação do robô, demonstração de cansaço ou não e atenção à tarefa. Os resultados indicaram um grande interesse das crianças pelos robôs e não medo em relação aos mesmos [1,12].

Verificou-se que ao utilizar o robô as crianças normalmente respondem melhor, em uma tarefa por períodos significativamente maiores, do que quando usado outros meios como programas gráficos de computador [13]. As crianças também pareceram entender as tarefas associadas a cada interruptor através de repetidas execuções [13]. Outra observação importante foi de que houve respostas positivas quando as crianças estavam olhando em direção ao manipulador robótico ou em direção ao interruptor. Um alto nível de atenção geral no momento da criança ativar o interruptor do robô foi possível de ser percebido [12]. Esse estudo complementou o trabalho anterior dos mesmos autores, uma vez que focou no entendimento da criança em relação ao seqüenciamento e uso do manipulador robótico para exploração.

Essa pesquisa feita com crianças com deficiências físicas severas pode também ser aplicada a crianças que possuem autismo, já que como resultado do próprio autismo, a criança se isola do mundo e tem uma interação limitada com o meio ambiente [14].

Assim, foram realizados diversos testes, relatados em [14], sobre como poderiam ser os robôs para ser comandados por tais crianças que possuem autismo, considerando fatores como velocidade de aproximação do robô e design do mesmo, que são itens importantes para a criança se sentir confortável e segura com o robô.

A conclusão da pesquisa com as crianças com autismo foi que o robô deve ser programado de acordo com o grau de dificuldade de interação com o mundo que a criança possui. Além disso, o robô deve ser visto como um brinquedo [15] que deixe a criança curiosa e interessada (e ao mesmo tempo segura), e se as tarefas começarem a ficar entediantes, as mesmas devem ser modificadas. É de grande importância que a criança tenha muita expectativa com o robô, por isso, tanto o comportamento como o design do mesmo devem ser interessantes para a criança.

O foco do trabalho com os autistas foi a interação com os robôs (o que ajuda, futuramente, com a melhoria na interação com as pessoas) e o estímulo da atenção em algo que está fora do “mundo criado pela criança autista” [14].

Já o objetivo com crianças com deficiência é de tirar, e também evitar, o surgimento da idéia de indefesa aprendida e dar-lhes mais independência de comunicação e expressão, diminuindo o isolamento das mesmas e melhorando a interação com o mundo.

Embora a abordagem seja diferente, a idéia de que as tarefas devem ser prazerosas e parecerem brincadeiras é comum tanto para as crianças autistas como para crianças com deficiência física. Quanto mais interesse houver pelos robôs, intui-se que melhor será o desenvolvimento das crianças nas tarefas propostas, pois sua atenção e empenho serão maiores.

Portanto, embora tenha abordagem diferente, algumas idéias dos trabalhos que tratam sobre o autismo podem ser utilizadas para as crianças com deficiência. Tais idéias, como tarefas não rotineiras e o fato do robô ser visto pela criança como um brinquedo divertido [14,15] são comuns a ambas as abordagens.

Para a realização da pesquisa desta dissertação de Mestrado, foi utilizado um robô móvel a rodas dotado de pinças de manipulação, o qual foi comandado por crianças com deficiência, utilizando alguns de seus sinais biomédicos voluntários.

As tarefas realizadas pelas crianças foram diversas, porém a idéia principal é fazê-las interagirem com o ambiente, pegando objetos e colocando-os em outros lugares, achando objetos “escondidos” e desenhando sobre um papel localizado sob o robô. Tarefas de percurso entre obstáculos no espaço seguindo critério de cores também foram realizadas.

Cada sessão foi gravada em vídeo para revisão (após assinatura, pelo responsável pela criança, do Termo de Consentimento aprovado pelo Comitê de Ética da UFES mostrado no Apêndice IV). Foram avaliadas as ações das crianças e seus comportamentos durante a execução das tarefas, e o tipo e número de comandos que são necessários para que se complete com sucesso a tarefa.

Tais fatores, como número de comandos certos e errados, tempo e a expressão da criança no momento em que foi feito o movimento, foram importantes para a avaliação posterior de desempenho da mesma.

1.2 Aspectos importantes a considerar

Todo este processo é visto pela criança como uma brincadeira [1,11,16] e, para isso, a mesma deve se sentir confortável, segura e ter motivação com o robô a ser utilizado. Ao levar este importante fator em conta é possível perceber que o design do robô é extremamente importante, bem como suas características se assemelharem a um brinquedo.

Cores, formas e comportamentos amigáveis são essenciais para que haja uma segurança para a criança que, ao utilizar o sistema, não terá medo do robô. Qualquer situação que apresente insegurança ou medo para a criança deve ser evitada, uma vez que a componente psicológica interfere de forma muito importante dentro deste processo de avaliação, como será explanado melhor nos capítulos seguintes.

Neste tipo de teste aplicado é de grande importância que a criança tenha um suporte pedagógico. Por isso Pedagogos acompanharam todo o processo para avaliar as questões psicológicas e educacionais envolvidas, o que pode interferir – e muito – nos resultados obtidos.

Além disso, as tarefas também foram de acordo com a idade e perfil da criança, já que isto influencia muito no desempenho dos testes. Algumas crianças são mais determinadas em atingir os objetivos das tarefas que outras [10] e, por isto, tais tarefas necessitam do suporte pedagógico.

Este trabalho está organizado da seguinte forma:

- No Capítulo 1 é feita esta introdução explicando em linhas gerais o problema a ser resolvido;
- O Capítulo 2 trata dos métodos de avaliação nos experimentos;
- O Capítulo 3 descreve o hardware utilizado;
- O Capítulo 4 explica o funcionamento do sistema como um todo;
- O Capítulo 5 apresenta como os experimentos serão realizados;
- O Capítulo 6 exhibe a abordagem pedagógica utilizada para avaliação;

- O Capítulo 7 mostra os resultados; e
- O Capítulo 8 apresenta as conclusões finais.

Os apêndices apresentam os seguintes conteúdos:

- Apêndice I mostra código do PIC;
- Apêndice II mostra código da interface no computador;
- Apêndice III apresenta as partes extras do robô;
- Apêndice IV a VIII apresentam os documentos do Comitê de Ética, incluindo a Carta de Aprovação.

Capítulo 2: Métodos para avaliação

Os resultados do processo de aprendizagem e da utilização do sistema pelas crianças com deficiência física severa devem ser mensurados para a percepção do desenvolvimento social e cognitivo da criança. Como todo esse processo envolve aspectos emocionais, é necessário o uso de um método de avaliação que leve em conta a estatística e outra que leve em conta o comportamento. Para isso, alguns autores sugerem que seja utilizada uma tabela com pontuações para avaliar comportamento, bem como, em paralelo, um método para medir o sucesso no cumprimento das tarefas. Este último método é chamado de GAS (Goal Attainment Scale ou Goal Accomplishment Scale [16] – em português, Escala de Objetivos Atingidos). Além disso, neste trabalho também são feitas entrevistas com professores para verificar a melhora cognitiva das crianças com o uso do robô [16].

2.1 Goal Attainment Scale (GAS)

Também conhecido como Goal Achievement Scale, este método utiliza pesos para os objetivos e usa uma nota para cada objetivo atingido. A pontuação de cada objetivo pode ser de -2 a +2, sendo que 0 é o resultado que já se esperava dos testes e pontuações positivas são resultados melhores que o esperado, enquanto pontuações negativas são resultados abaixo do esperado. Vale à pena mencionar que esta escala possui um alto grau de subjetividade, pois leva em conta qual é o tipo de deficiência motora que a criança possui e deve ser analisada junto com a tabela de comportamento que será explicada na seção 2.2.

Por fim, a pontuação geral é calculada utilizando as notas de todos os objetivos atingidos pelas crianças e dando um peso para cada um. A equação (1) representa a pontuação geral (*overall score*) [15]:

$$T = 50 + 10 \cdot \left(\frac{\sum_{i=1}^n g_i}{\sqrt{n - R \cdot n + R \cdot n^2}} \right) \quad (1)$$

Onde

- g_i – pontuação dada pela meta i a ser alcançada.
- n – número de objetivos a ser alcançados na tarefa (uma tarefa pode ter vários objetivos e, além disso, a realização parcial deles também é levada em conta).
- R – constante usada para estimar a correlação entre a pontuação entre os múltiplos objetivos nas tarefas. Na literatura comumente é utilizada a constante 0,3 [15].

A equação (1) deve ser utilizada de forma comparativa, segundo [16], pois se deve avaliar o ganho de pontuação geral antes do treinamento com os robôs (a primeira vez que realizam as tarefas) e após sucessivos testes utilizando os robôs. Desta forma, avalia-se quanto se ganhou em termos de aprendizado e pontuação (que mede quanto de objetivos a mais foram alcançados com a aprendizagem de como utilizar o sistema), sendo esta uma ferramenta quantitativa para mensurar o desenvolvimento cognitivo.

2.2 Medição com base no comportamento

É notável que sob certas emoções como medo e tristeza os resultados tendem a ser piores, enquanto atenção e motivação fazem com que os resultados sejam melhores. A medição desta escala utiliza certos comportamentos para verificar se a criança está prestando atenção à tarefa ou está entediada. Isso é importante e deve ser levado em conta, pois interfere de forma substancial no desempenho da mesma.

Com isso, é importante observar que o GAS e o comportamento possuem uma correlação, já que crianças com humor melhor conseguem atingir os seus objetivos mais facilmente do que aquelas entediadas. Porém, vale a pena citar que esses dados são subjetivos e podem ser avaliados por profissionais da área de Educação, enquanto o GAS (embora tenha um grau de subjetividade) pode ser considerado uma forma estatística de avaliação.

Capítulo 3: Hardware

O hardware utilizado neste trabalho compõe-se basicamente de um computador, um sensor inercial, um sensor mioelétrico e um robô, todos capazes de realizar conexão Bluetooth. Os sensores são responsáveis por captar os movimentos gerados pela criança, através de um sensor de inclinação (baseado em acelerômetro) e esforço muscular realizado (através de um sensor mioelétrico sEMG de um canal).

3.1 Robô Móvel com Manipulador

O robô móvel a rodas utilizado nesta pesquisa é o POB-BOT (com a placa POB-EYE), fabricado pela POB TECHNOLOGY [17]. Trata-se de um robô móvel a esteiras dotado de pinças, o que permite trabalhar com o mesmo de duas formas, como robô móvel e como manipulador.

A figura 3 mostra o robô que foi utilizado nos experimentos. Ressalta-se que foi acoplado ao robô um dispositivo de comunicação por Bluetooth e um enfeite em forma de palhaço, que serão explicados posteriormente.

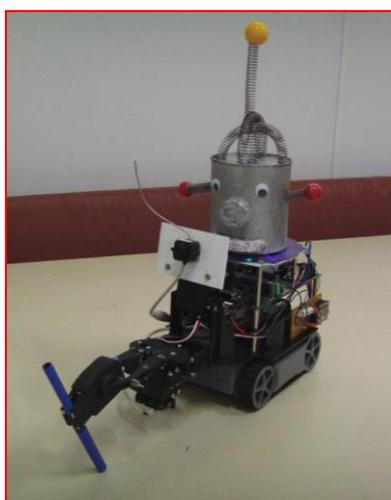


Figura 3 - Foto do robô POB-EYE com Bluetooth conectado (parte lateral).

Este robô possui três servos-motores para realizar os movimentos de pinça (levantar e abaixar), de garra (abrir e fechar) e movimento da região onde está localizada a câmera [17]. Esses motores dão a características de robô manipulador ao POB-BOT.

Em conjunto a esta parte de manipulação do robô, está acoplada a parte móvel do mesmo, que conta com dois motores independentes CC de 12V. O controle e o sentido de rotação destes motores é feito por um circuito LM293D que possui uma dupla ponte H, o que permite a movimentação das esteiras do robô para frente, trás, direita e esquerda através da combinação de movimentação dos motores. As placas do POB-BOT que são utilizadas serão descritas a seguir enquanto que as que não foram usadas no projeto, porém fazem parte do robô, estão explicadas no Apêndice II.

O modelo cinemático do robô está representado pela figura 4 e equações (2) e (3) e é uma adaptação de [18].

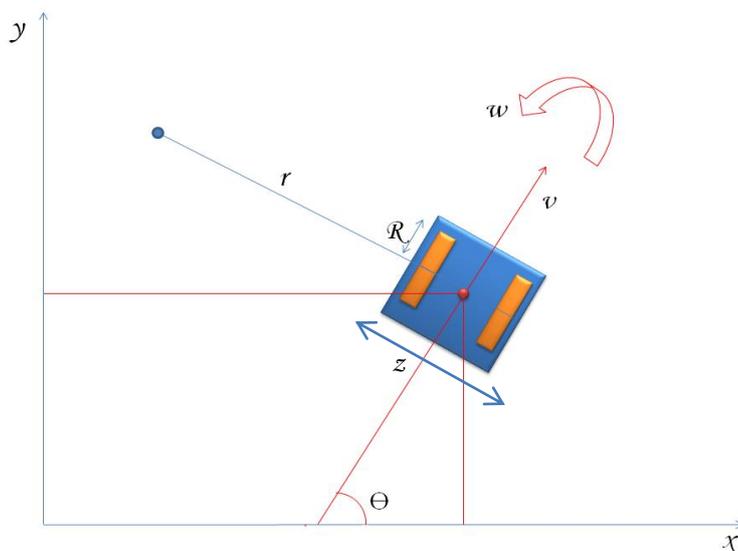


Figura 4 - Robô no plano xy .

Considerando as variáveis com índice ‘d’ sendo relativas à esteira direita e, analogamente, as com índice ‘e’ relativas à esteira esquerda, temos a equação (2):

$$\begin{aligned} v_d &= \omega_d R \\ v_e &= \omega_e R \end{aligned} \quad (2)$$

Portanto, a velocidade resultante do robô é (considerando o sentido anti-horário como positivo e a referência sempre sendo a parte da frente do robô). Isso é mostrado na equação (3):

$$v = v_d - v_e \quad (3)$$

O movimento angular do robô é dado pela equação 4:

$$\omega = \frac{v}{r} \quad (4)$$

Sendo que ‘r’ é o eixo no qual o robô está girando e não o raio das rodas que estão envoltas pela esteira.

Além disso, as posições x e y podem ser escritas de acordo com a velocidade linear como mostradas na equação (5):

$$\begin{aligned} \dot{x} &= v \cdot \cos(\theta) \\ \dot{y} &= v \cdot \sin(\theta) \\ \dot{\theta} &= \omega \end{aligned} \quad (5)$$

Com isto e considerando ‘z’ a distância entre as rodas direita e esquerda pode-se escrever a equação (6):

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} R/2 & R/2 \\ R/z & -R/z \end{bmatrix} \cdot \begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix} \quad (6)$$

Quanto à posição do robô pode-se afirmar a equação (7):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \dot{\omega} \end{bmatrix} \quad (7)$$

3.1.1 POB-PROTO – A placa de controle

O controle de todo o robô está na placa POB-PROTO, que comunica com as placas POB-LCD e POB-EYE, que serão citadas posteriormente. Todos os comandos da placa POB-PROTO são processados por um PIC16F877A [17]. A figura 4 mostra os componentes presentes nesta placa de controle central do robô.

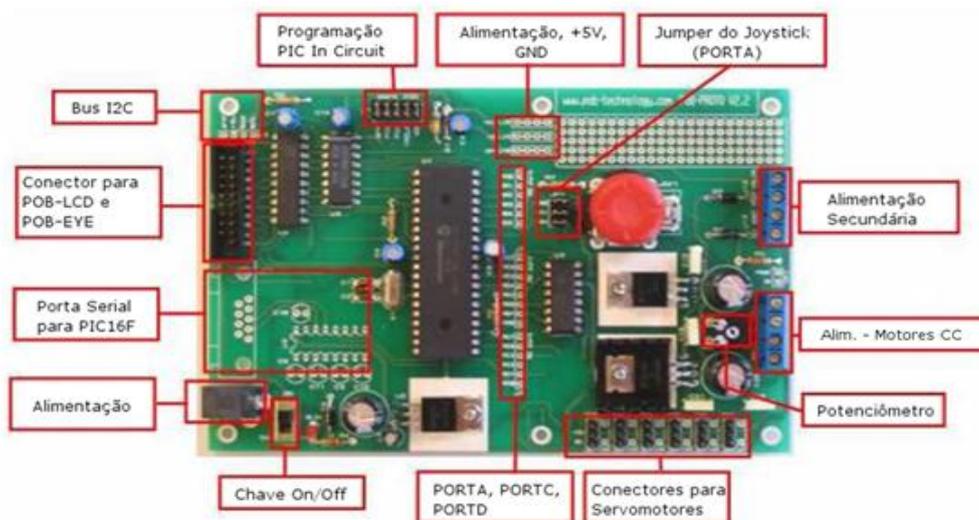


Figura 5 - Placa POB-PROTO [16].

Como dados técnicos do robô tem-se que a tensão de alimentação dos motores CC são de 12V e dos servos são de +5V, fornecidas pela alimentação principal. Além disso, na parte superior existe uma região com vários conectores com tensão de alimentação de +5V e GND. A tensão de alimentação deve ser entre 9 e 12V.

A programação feita no microprocessador está descrita no Capítulo 5 e o código do programa está no Apêndice III.

3.1.2 Placa de recepção Bluetooth

Para uma interface de comunicação sem fio do robô com o computador foi necessário utilizar uma placa Bluetooth desenvolvida anteriormente [19], a qual se conecta aos pinos C6 e C7 do PIC, que são respectivamente o RX e o TX. O módulo Bluetooth utilizado foi o da Roving Networks, que trabalha com taxa de transferência de até 3 Mbps [20]. Na figura 5 pode-se ver uma imagem desta placa (o módulo Bluetooth é a peça maior).



Figura 6 - Placa de recepção Bluetooth.

O circuito eletrônico da placa de recepção está mostrado na figura 6.

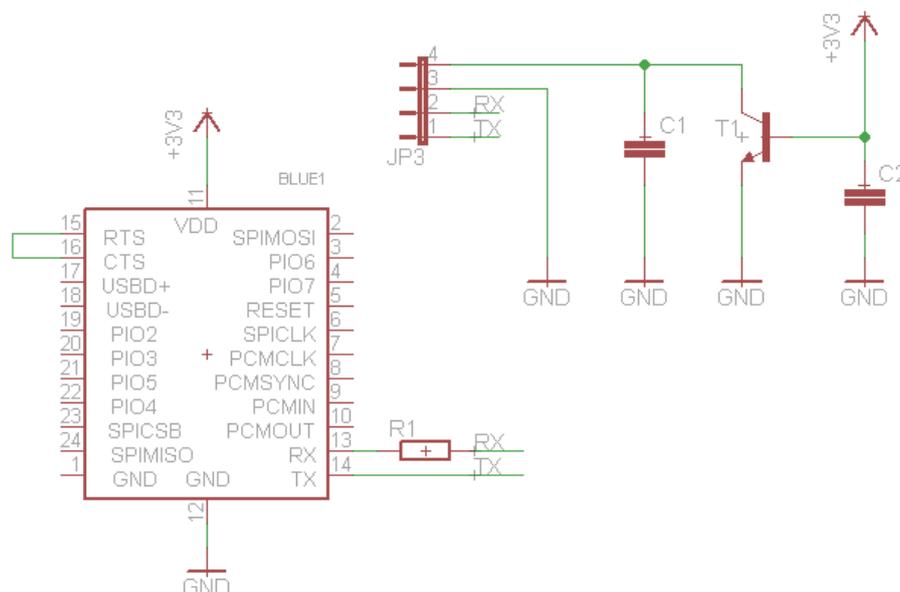


Figura 7 - Diagrama esquemático da placa de recepção [17].

O sistema utiliza este dispositivo como uma porta serial SPP com taxa de transmissão de dados de 115.200 bps no Windows, sendo que o computador o reconhece como uma porta COM. Apesar da comunicação entre os dispositivos se dar em 115.200 bps vale a pena ressaltar que o dispositivo Bluetooth permite através do taxas de até 3Mbps. No entanto, no computador, foi utilizada as portas COM padrão do Windows, pois a velocidade com que o sensor manda os dados era sempre igual ou inferior à 115.200 bps, não havendo necessidade de utilizar capacidade extra da porta USB que estava com o Bluetooth anexado.

3.1.3 Alterações realizadas no robô

O robô possui algumas placas extras, porém estas não serão utilizadas (os detalhes destas placas não utilizadas estão descritas no Apêndice III). O LCD disponível no robô não foi utilizado devido a problemas de compatibilidade com o sistema operacional do computador (Windows 7 32bits) e, da mesma maneira, a comunicação I2C existente no robô para capturar imagens da câmera tampouco foi utilizada.

O LCD foi substituído por um enfeite para chamar mais a atenção das crianças (mostrado na figura 7) e a câmera foi trocada por uma micro-câmera de 1,2 GHz para

um receptor de TV que está conectado a um dispositivo que converte as imagens para o padrão USB para serem adquiridas facilmente em qualquer computador.

Desta forma a única placa do robô original usada é a POB-PROTO, a qual controla os motores e servos-motores do robô e faz a comunicação com o computador.



Figura 8 – Componentes extras - câmera de segurança sem fio e conversor de sinal de televisão para entrada USB [21], e o enfeite utilizado no robô.

3.2 Placa de aquisição de sinais

Para a utilização deste robô nos experimentos, é necessário o uso de uma placa auxiliar que captura os sinais mioelétricos (sEMG), juntamente com o sinal de um sensor de inclinação baseado em acelerômetro. Tais sinais são capturados pelo sensor desenvolvido e enviados via Bluetooth para um computador, sendo que tal sensor é mostrado na figura 8.

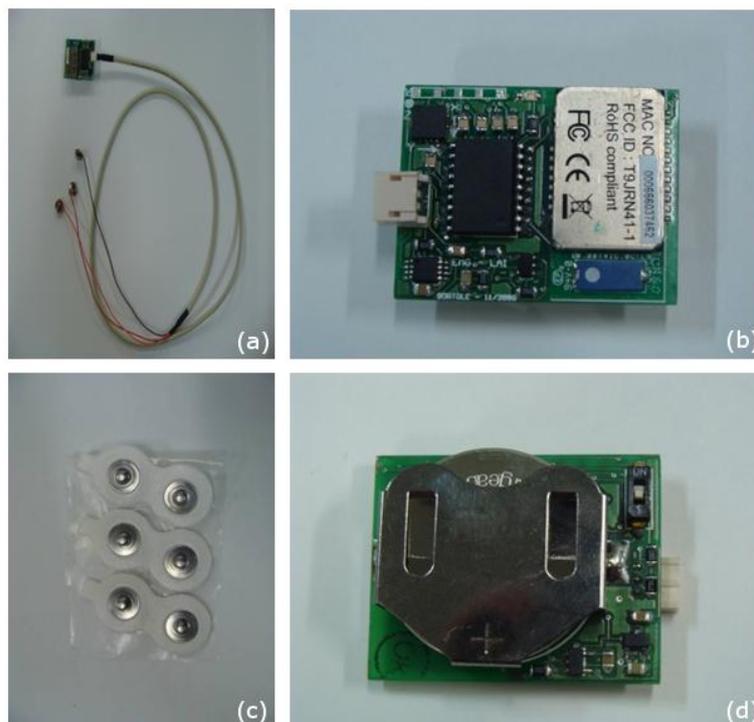


Figura 9 – Sensores - (a) Cabo de conexão para sensores sEMG, (c) eletrodos de superfície, (b) parte frontal e (d) parte posterior da placa de aquisição de sinais de inclinação e sEMG [19].

O processo que ocorre dentro deste circuito é uma conversão A/D de 10 bits que transforma a inclinação da cabeça nos três planos e captura o valor de tensão correspondente, que por fim é convertido em um valor digital através do microcontrolador PIC integrado ao circuito.

3.3 Câmera Externa

Uma câmera externa foi utilizada para visualizar o rosto da criança no momento em que são feitos os movimentos. Isso é importante para verificar qual é a reação da criança durante a tarefa. São capturadas imagens da face da criança a cada vez que é feito um movimento novo. Estas partes são colocadas nos relatórios para posterior análise por um Pedagogo.

A câmera utilizada foi uma *webcam*, modelo C3 Tech (mostrada na figura 9) de 300 kilopixels, podendo chegar a 16 megapixels interpolados por software.



Figura 10 - Webcam para tirar fotos das crianças durante a realização dos experimentos [22].

Capítulo 4: Funcionamento do Sistema

O sistema possui quatro etapas de funcionamento que podem ser separadas em:

- 1) Aquisição de sinais e conversão pelo dispositivo de captura;
- 2) Transmissão via Bluetooth dos dados;
- 3) Interpretação dos dados pelo computador e geração dos relatórios;
- 4) Envio para o robô e movimentação do mesmo.

Cada uma das quatro etapas será explicada em detalhes a seguir, sendo que a figura 10 representa o processo como um todo. Nela é possível verificar que inicialmente há a captura dos sinais através dos sensores. Estes sinais são então digitalizados e enviados via Bluetooth para um computador que os interpreta e, por fim, envia o comando de movimentação correto ao robô enquanto, em paralelo, gera o relatório da criança.

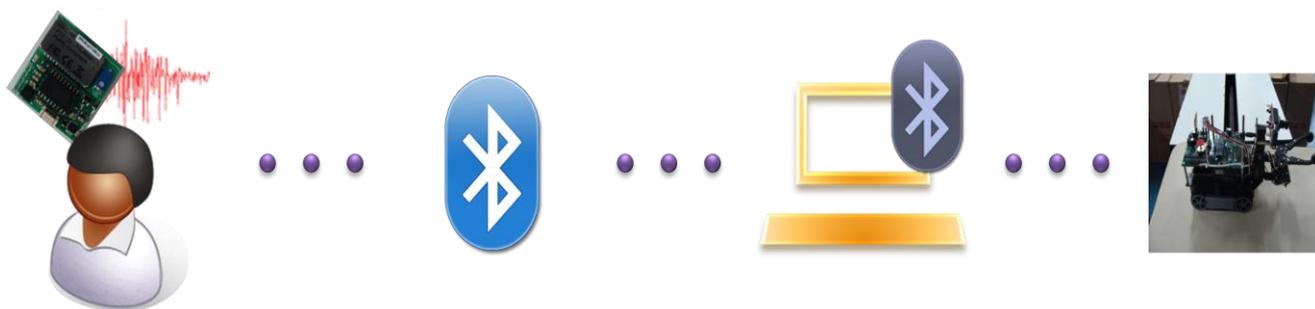


Figura 11 - Esquema utilizado para captura dos sinais e movimentação do robô

4.1 Parte 1: Aquisição e conversão dos sinais

O dispositivo de captura de sinais citado anteriormente pode ser colocado em um boné na cabeça da criança ou no seu pulso com uma pulseira de pano, de tal forma que é

possível colocar o sensor de forma confortável para a criança utilizar. Para prender melhor a placa, utilizam-se velcros. Na figura 11 pode-se visualizar o boné, a pulseira e o circuito.



Figura 12 - Sensor junto com os suportes utilizados nos testes.

Sentada em uma posição confortável, de tal forma que possa ter visualização do robô, a criança veste o boné para trás (para a viseira não atrapalhar a visão). A criança, então, recebe instruções sobre qual experimento será feito e como o mesmo será realizado. Por fim, a interface é configurada, sendo inseridos os dados relevantes para o processo.

A aquisição dos sinais pela placa se inicia a partir do momento em que a mesma é ligada, porém o computador só começa a receber e ler as informações recebidas no momento em que é pressionado o botão verde (iniciar tarefa).

4.2 Parte 2: Transmissão de dados via Bluetooth

A partir do momento em que na interface é clicado o botão verde (começar tarefa), a porta COM configurada previamente é lida de forma contínua e os dados são interpretados pela interface. Lá os sinais são separados e transformados de *strings* para *integers* de 4 algarismos. Para identificar o início de um pacote (e automaticamente o término do anterior), é utilizado o caractere *i*, como mostrado na figura 12.

A transmissão de um conjunto completo de informações do sensor sobre a inclinação da cabeça da criança, o esforço muscular exercido (sinais SEMG), o nível de bateria e início do pacote se dá em 94 bytes, sendo que o pacote começa com o caractere 'i' [19]. São 20 sinais sEMG (representando 80 bytes), 3 sinais de inclinação (um em cada plano representando 12 bytes no total) e 1 sinal de bateria com 1 byte.

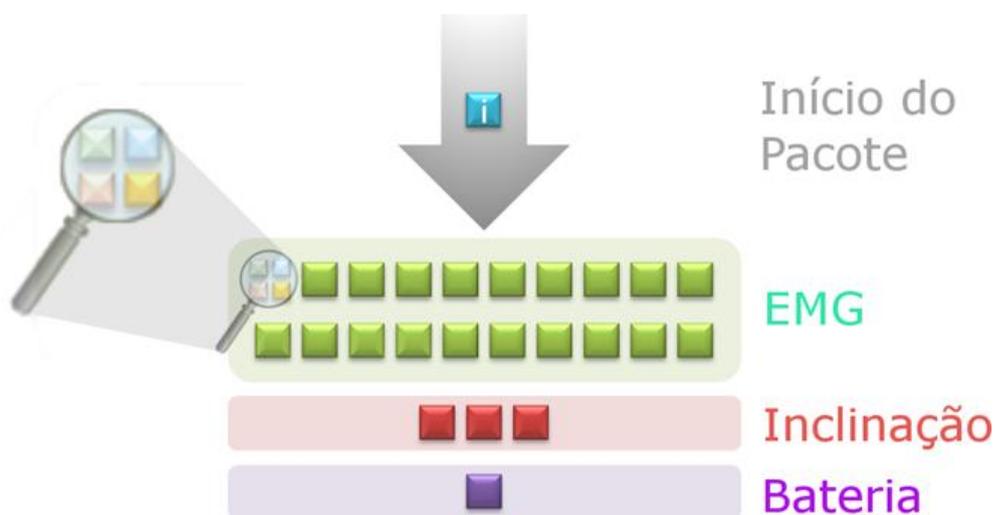


Figura 13 - Forma como as informações são enviadas pelo sensor. Observe que cada sinal tem quatro bytes, representados em destaque dentro da lupa.

4.3 Parte 3: Interface

A interface é a parte do sistema que faz a ponte entre o robô e o sensor que está no boné da criança. Consiste basicamente em guardar cadastros, converter os dados, tirar as fotos da criança quando fazem movimentos com o robô, gravar os movimentos feitos, e gerar relatórios.

Esta interface foi desenvolvida utilizando o Microsoft Visual Basic Express Edition 2008, utilizando recursos da plataforma .NET.

4.3.1 Interface – Funcionamento geral

O software desenvolvido faz a parte de coleta dos sinais já convertidos para o formato digital, bem como traduz estes movimentos para o robô. A programação feita pela interface permite que qualquer dispositivo de captura seja utilizado com qualquer robô, desde que existam os comandos equivalentes nos dois equipamentos. No entanto, para o hardware de captura citado anteriormente já existe uma configuração padrão já pronta dentro do código. Isso pode ser visto com mais detalhes no apêndice I.

As principais funções da interface consistem em:

- Coletar os dados vindos da placa de captura;
- Converter estes dados para o robô selecionado;
- Enviar os dados para o robô;
- Captura foto da criança e gerar relatório.

Desta forma, de acordo com o movimento feito pela criança, o robô pode comandar o robô para realizar os movimentos de ir para frente, para trás, para direita e para a esquerda. Para fazer o movimento das pinças, é necessário utilizar sinal mioelétrico, que é captado com eletrodos ligados à pele da criança para obter o sinal

elétrico do músculo, sendo utilizada a contração muscular para mudar o estado em que a garra se encontra; se estiver para baixo, ela sobe, e vice-versa.

Em contrapartida, o sinal do acelerômetro é recebido em todo momento e seu valor é convertido para um movimento para o robô. São verificados dois valores, um máximo e um mínimo, para cada um dos eixos de inclinação.

Com esses valores em mão e sabendo que o valor máximo para o eixo X vai para trás e o valor mínimo do eixo X vai para frente (e, de forma análoga, no Y o valor máximo vai para esquerda e o valor mínimo vai para direita), pode-se determinar faixas de valores nos quais o robô irá se movimentar. A figura 13 representa graficamente os movimentos de acordo com o eixo.

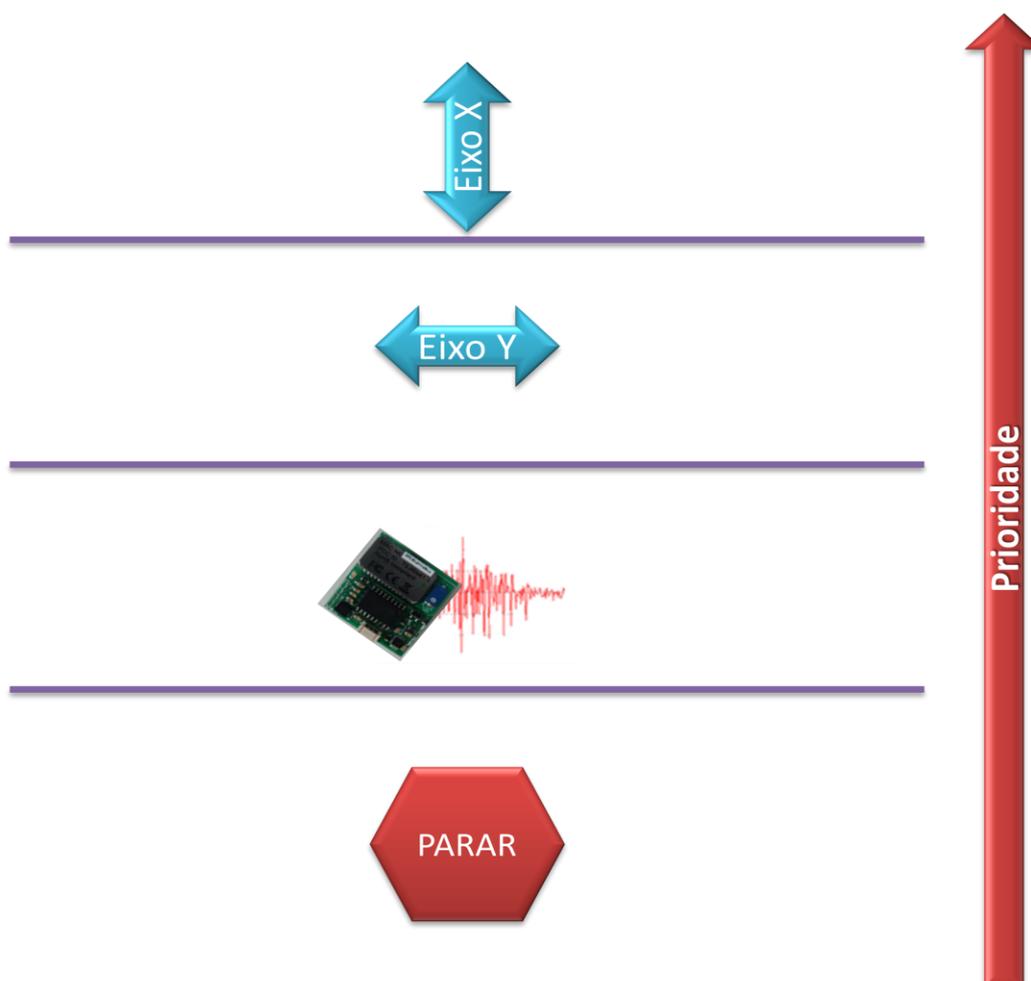


Figura 14 - Prioridade dos eixos.

Como se pode ver na figura, o eixo X predomina sobre o Y, isto é, se chegar valores de X e Y em faixas que o robô deve se movimentar, o mesmo irá seguir o comando do eixo X. Em outras palavras, o robô analisa primeiro se deve ir para frente ou para trás e caso nenhuma dessas opções seja válida, analisa se deve ir para a esquerda ou para a direita.

Os valores podem ser pré-definidos ou obtidos na janela de calibração. Esta janela indica que a criança deve ficar 10 segundos na posição de repouso e então captura o sinal de inclinação ou sEMG correspondente e grava qual é o comando relacionado àquele sinal biomédico. No tópico 5.8 será explicado melhor o processo de calibração.

Por fim, para encerrar o ciclo, o computador converte os dados para comandos compreensíveis ao robô. No caso da programação para o robô padrão (POB-BOT), os comandos 'w', 's', 'd' e 'a' significam para frente, trás, direita e esquerda, respectivamente. Em terceiro nível de prioridade existem os comandos de troca de estado da pinça e garra, os quais são interpretados pelo robô pelas letras 'p' e 'g' respectivamente, que são ativados através do sinal mioelétrico.

Graficamente esta disposição de movimentos pode ser representada pela figura 14 onde o ponto central verde é o estado em que o robô fica parado.

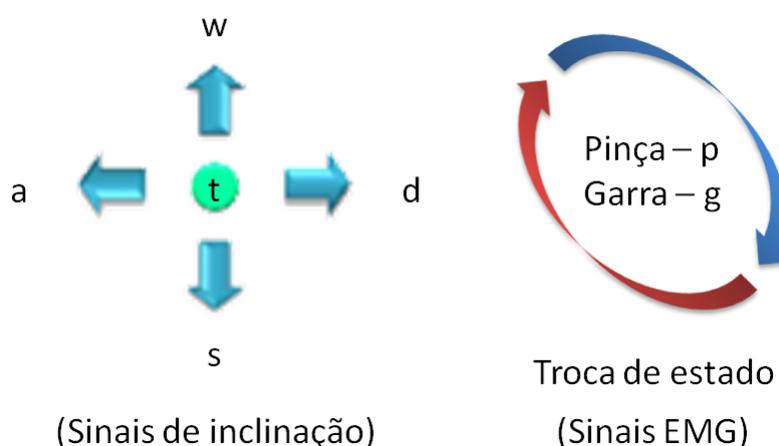


Figura 15 - Caracteres que representam cada movimento do robô. Ao receber estes caracteres o robô se movimenta de acordo com a imagem correspondente.

Quando a placa fica em uma posição em que não atenda a nenhum dos movimentos, é enviado o sinal ‘t’ para robô, o qual comanda o mesmo para ficar parado. Se este sinal não for enviado, o robô se movimenta até recebê-lo ou receber outro sinal de movimentação.

A mesma idéia é utilizada para o sinal sEMG, porém, este tem apenas uma componente e existe um valor definido, no qual acima dele o sinal é considerado como válido para movimentar a pinça.

Com essas idéias e lembrando que o robô está em cima de um papel para desenho, a criança tem capacidade de controlá-lo para desenhar algo. Para rabiscar o desenho no papel o robô tem adaptado à sua garra um pincel.

Vale a pena lembrar que para a tarefa de desenhar, outro comando, representado pela letra ‘g’, que permite trocar o estado de abertura da garra, entre fechar e abrir, está travado via software. Isso é necessário devido à presença de um suporte para fixar o pincel, porém, em outras atividades como pegar caixas e objetos pequenos, este movimento é habilitado.

4.4 Parte 4: Interface – Organização dos dados

Além do funcionamento geral a interface também é responsável por organizar os dados e guardar os relatórios, funcionando assim como um banco de dados. De forma geral a interface tem seis telas:

- Tarefas principais
- Adição de tarefas
- Relatórios
- Cadastro de pessoas
- Cadastro de robôs
- Cadastro de dispositivos auxiliares

As seis partes do programa se relacionam conforme a figura 15. Como entradas de dados têm-se todas as partes do programa, com exceção da geração dos relatórios, que é a saída do sistema e, que de fato, é o resultado apresentado pelo mesmo.

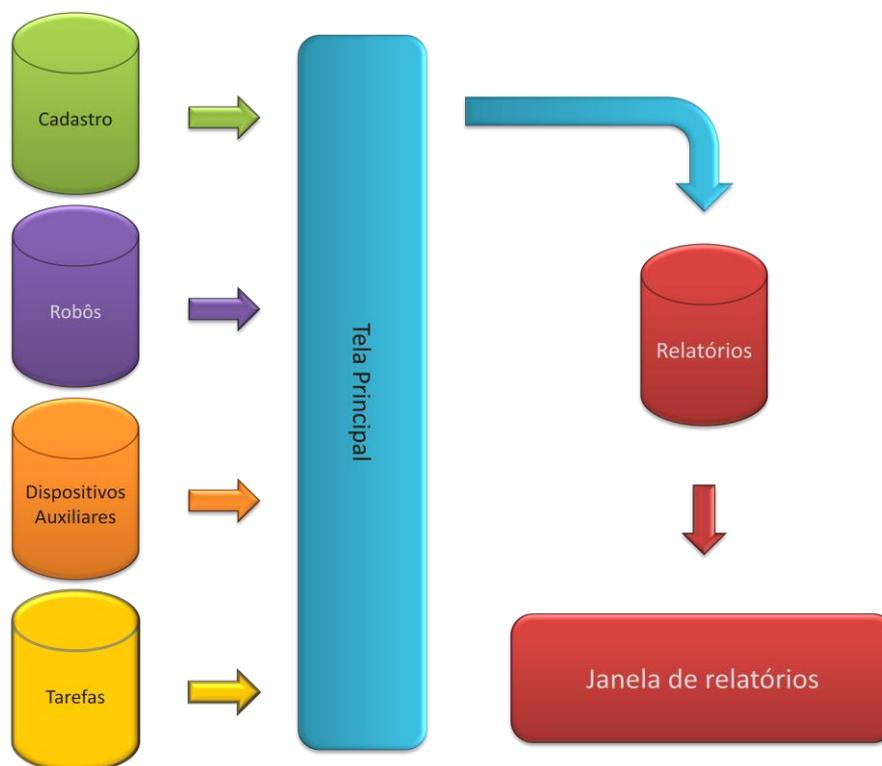


Figura 16 - Esquema simplificado de como são gerados os relatórios de testes.

Como suporte à parte principal do programa existem as telas de cadastros, tanto de pessoas quanto robôs e dispositivos auxiliares.

Esses cadastros funcionam de forma parecida, conectados a um banco de dados SQL Server Compact, presente no pacote de instalação do Visual Studio Express Edition 2008. Com essas informações no banco de dados é possível fazer o relatório, após os testes, pois toda informação a ser coletada está presente lá. Além disso, se for necessário recuperar alguma informação ou fazer estatística de algo pode-se recorrer ao banco de dados que pode ser lido tanto pela interface desenvolvida quanto por qualquer programa capaz de ler e alterar banco de dados.

Cada cadastro possui seu próprio banco de dados e tabelas. Em mais detalhes, isso pode ser representado graficamente pela figura 16.

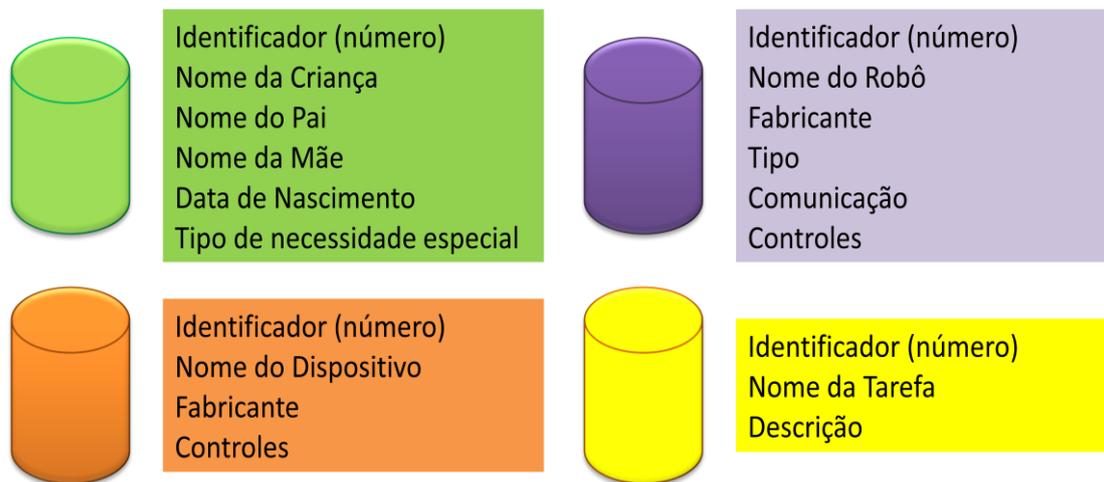


Figura 17 - Representação gráfica de quais informações são guardadas em cada um dos bancos de dados.

Além disso, é importante citar o banco de dados de relatórios, os quais são utilizados para gerar os arquivos HTML que são lidos pelos browsers. Este banco de dados possui elementos dos outros e está representado pela figura 17:



Figura 18 - Representação gráfica das informações guardadas no banco de dados de relatórios utilizado para gerar os arquivos HTML.

A tela mais importante de todo o software é, sem dúvida, a janela de tarefas principais, pois, além de ser possível acessar quase todas as outras janelas a partir dela é com essa tela que se comanda todo o processo dos experimentos. Na figura 18 pode-se visualizar a tela principal.



Figura 19 - Tela principal do software de robótica educacional.

Nesta parte do programa é possível selecionar o nome da criança, que já está previamente cadastrado em um banco de dados, bem como a tarefa a ser realizada e qual robô será utilizado para isso. Tais informações são importantes, pois serão utilizadas posteriormente na confecção do relatório.

Questões de conexão entre o sensor (acelerômetro e sEMG) e o robô também são resolvidas nesta tela, pois se pode selecionar a quais portas estão conectados tais dispositivos e também pode definir se deseja apenas trabalhar com o sistema utilizando o controle alternativo presente na tela. Para isso basta não conectar o sensor e o robô.

Além do controle alternativo existe uma visualização alternativa do robô, representada de forma simplificada pelo ponto amarelo dentro do espaço azulado. Isso é apenas necessário para efeitos de auxílio à tarefa, quando alguém está longe do robô e

não pode vê-lo ou quando o robô está desligado e então se trabalha apenas com este “robô virtual”.

Na parte da tela de conexão entre o robô e o sensor existe um indicador que, se ficar verde, significa que foi possível conectar o dispositivo e, caso fique vermelho, significa que o dispositivo não conseguiu se conectar.

O esquema representado pela figura 19 mostra, de forma simplificada, como funciona o programa nesta tela.

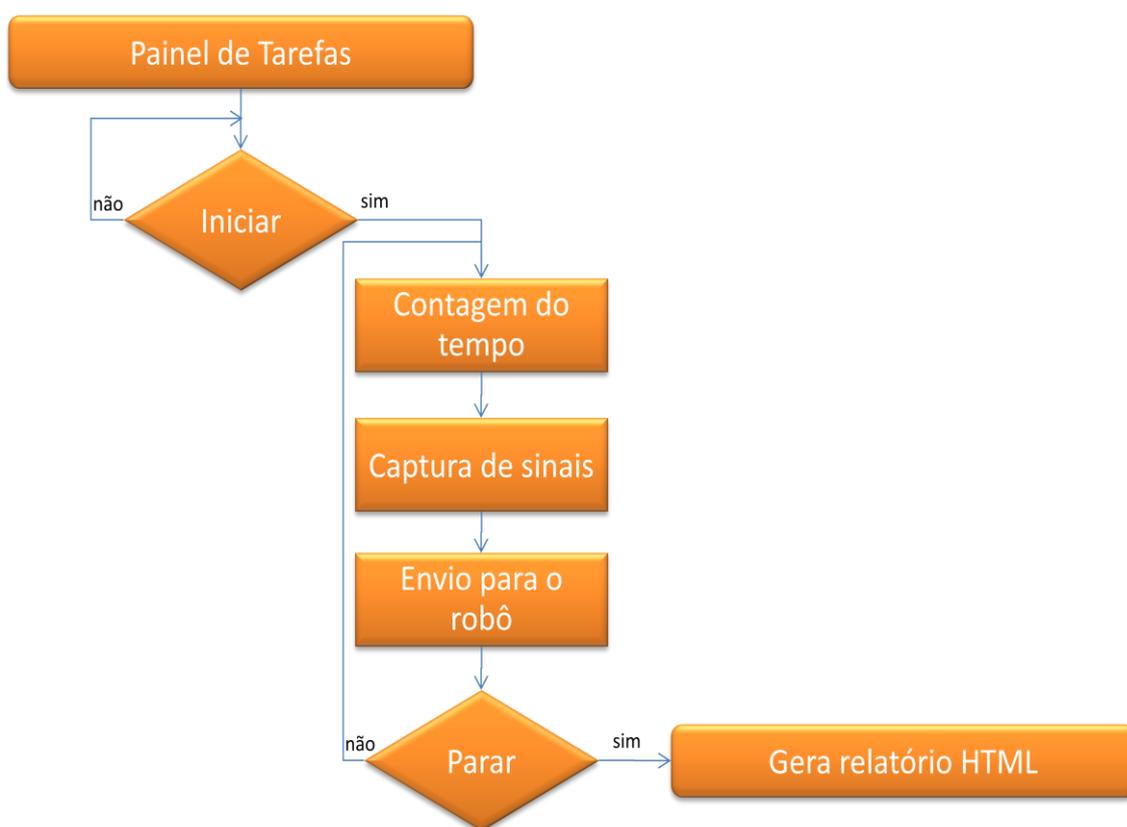


Figura 20 - Esquema simplificado de funcionamento do sistema.

Vale a pena citar que este esquema já considera que foram selecionados uma criança, um experimento e um robô, já contidos no banco de dados. Além disso, supõe-se também que uma câmera já tenha sido selecionada. Dentro do comando “envio para o robô” há a análise se o movimento é correto, errado ou livre e, após isso, o movimento é gravado no banco de dados e, por fim, é enviado para a porta Bluetooth que comanda o robô para se movimentar.

Outro ponto importante é que as palavras “iniciar” e “parar” representam, respectivamente, o botão que está entre as setas, no *groupbox* de controle, no modo “iniciar” e “parar” (figura 19). Ainda dentro desta tela existe um link para “seqüência correta”, que permite gravar a movimentação correta do robô, de tal forma que o mesmo só se mexe com a movimentação correta do sensor.

4.5 Cadastros e conversão de dados

No cadastro de pessoas são perguntados apenas o nome da criança, data de nascimento, qual tipo de deficiência da criança e o nome dos pais. Já no cadastro de tarefas pergunta-se apenas o nome da tarefa e também uma breve descrição da mesma. Em ambos os casos cada pessoa ou tarefa possui um número único de identificação, que serve para facilitar nas buscas e organização do programa em termos de código.

Quanto ao cadastro de robôs e dispositivos, alguns dados a mais, e de grande importância, são perguntados. Além do nome e do fabricante, pergunta-se o tipo e como é que o sensor envia o comando de ir para frente, trás, direita e esquerda, além dos comandos dos botões especiais X, Y, A, B e C, presentes na tela da tarefa principal.

Como isso é feito tanto no cadastro de robôs como no cadastro de dispositivos, é possível ser feita uma tradução entre os sinais enviados pelos dispositivos e os comandos entendidos pelo robô. O responsável pela tradução é um sistema intermediador que, neste caso, é o próprio software de robótica educacional. Uma representação gráfica disto é exibida na figura 20.

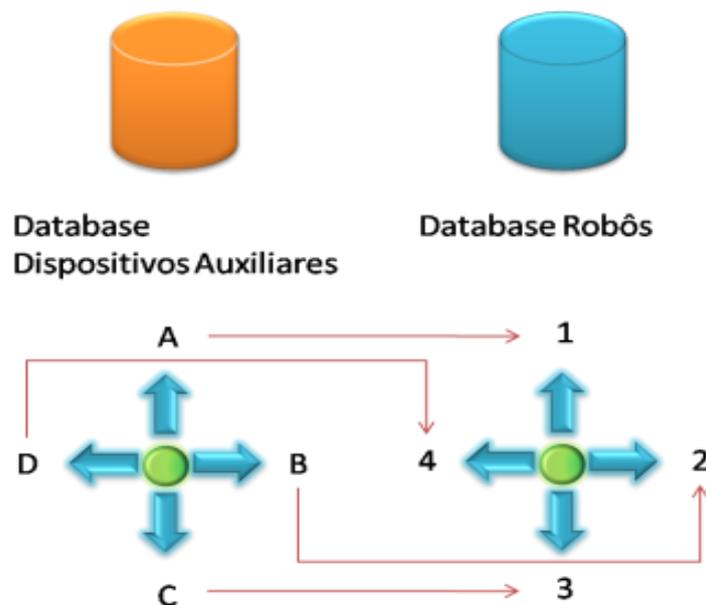


Figura 21 – Tradução - a conversão é realizada comparando itens correlatos nos bancos de dados.

Com esse sistema de tradução é possível cadastrar como um sensor envia os sinais de movimento e como o robô os entende. Tais cadastros são feitos de forma independente e, no momento em que se é selecionado o sensor e o robô, na janela de tarefas, esse esquema de tradução já é internamente montado.

Para exemplificar melhor essa situação, pode-se ilustrar a situação em que há um dispositivo auxiliar (sensor), como por exemplo, um acelerômetro, instalado em um boné na cabeça da criança, e a mesma pode comandar vários tipos de robôs. Da mesma forma, pode-se trocar o sensor por um dispositivo que captura, por exemplo, sinais EEG ou sEMG e a criança também controlar os robôs, porém de uma maneira diferente.

4.6 Tratamento do sinal de sEMG

Existem vários métodos para o tratamento do sinal sEMG e a calibração dos dispositivos. Alguns pesquisadores utilizam a envoltória do sinal e calcula seu valor médio, outros fazem análises estatísticas e têm os que analisam o tempo da onda em que o esforço está “ativo”. Esse tratamento do sinal muscular é muito importante, uma vez

que os sensores de forma geral são suscetíveis a ruídos e podem considerar um sinal como sendo um sinal mioelétrico, porém sem o ser.

Para resolver este problema, foi estabelecido um limiar (*threshold*) para que seja necessário que o sistema considere o sinal como realmente sendo um sinal muscular. Porém, ainda assim, um ruído muito alto pode fazer com que haja uma interpretação errônea da onda. Então, aproveitando as 20 amostras por ciclo de leitura feito pelo dispositivo de captura é feita uma média de todas as amostras e, então, se a média RMS estiver acima do *threshold* considera-se a movimentação como válida (equação 8).

$$\text{Media (RMS) Emg} = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (\text{Emg}_i)^2} \rightarrow \begin{cases} \text{Media Emg} \geq \text{threshold} \rightarrow \text{sinal válido} \\ \text{Media Emg} < \text{threshold} \rightarrow \text{sinal inválido} \end{cases} \quad (8)$$

4.7 Calibração do sensor de inclinação

Para o sensor inercial em específico há uma parte do software que permite a calibração do mesmo, de tal maneira que o próprio programa captura os valores da conversão AD relativa a cada movimentação. Vale a pena ressaltar que esta parte do programa é válida apenas para este sensor.

O funcionamento básico desta parte do programa se resume em pedir que a criança coloque-se na posição em que ela queira que o robô vá para direita, esquerda, frente e trás e então se se pressiona o botão correspondente na tela. Durante 10 segundos são capturados os valores e então é tirada uma média. Esta média é utilizada como valor-base para movimentação naquela direção.

Em outras palavras, ao atingir um valor maior que 90% do valor-base para a movimentação para esquerda ou para trás, o robô já começa a se movimentar. No caso da direita e frente o valor tende a cair, portanto, se chegar a ficar menor que 110% do valor-base já começa a se movimentar. Isso se for utilizado uma tolerância de 10%, porém pode-se mudar a mesma, de acordo com os movimentos possíveis para cada criança.

Da mesma forma que para a validação do movimento via sEMG, a calibração do sensor de inclinação é dada pela equação (9).

$$\text{Inclinação} \rightarrow \begin{cases} \text{Inclinação} \geq (1 - \text{tolerância}) * \text{threshold}_{\text{máximo}} & \rightarrow \text{trás/esquerda} \\ \text{Inclinação} \leq (1 + \text{tolerância}) * \text{threshold}_{\text{mínimo}} & \rightarrow \text{frente/direita} \end{cases} \quad (9)$$

Vale à pena citar que valores fora das faixas acima descritas caracterizam o movimento como “parado” e qualquer movimento em execução é interrompido. Na figura 21 pode-se visualizar a tela de calibração.



Figura 22 - Tela responsável pelas calibrações.

4.8 Modos possíveis de operação

Como mostrado na seção 5.1 a interface tem suporte para conexão entre o robô e um dispositivo auxiliar, nesta tela chamado de sensor, através de portas seriais. Apesar da capacidade de se conectar a dispositivos reais, não é necessário nem que haja um robô nem um sensor conectado para que a interface funcione. Tais modos não têm função para a pesquisa e coleta de dados com as crianças, porém é útil para detectar problemas no funcionamento do sistema como um todo. No total são quatro os modos possíveis de funcionamento do sistema:

- **Sistema totalmente virtual:** neste modo não se conecta nem o robô nem os sensores. O controle lateral é utilizado para comandar o movimento do robô virtual representado pela bolinha amarela ao lado esquerdo da tela.
- **Apenas com conexão com robô:** neste modo pode-se ver o funcionamento do robô separado do funcionamento do sensor. Não há conexão com o sensor, sendo que o robô (real) conectado via Bluetooth é comandado através dos controles laterais.
- **Apenas com conexão com o dispositivo auxiliar:** o robô virtual é acionado através de movimentos reais neste modo, sendo que é o único além do modo sistema completo que pode ser útil para pesquisa, já que capta os movimentos reais da criança.
- **Sistema completo:** este será o modo utilizado nas pesquisas com as crianças. Ele captura os dados do sensor, processa o relatório e move o robô virtual e real. Deve haver conexão entre o sensor e o robô para entrar neste modo.

4.9 Relatórios automáticos

A interface desenvolvida é capaz, ao final da tarefa, de gerar um relatório contendo todos os dados relevantes do experimento. Isso é útil para organizar as informações para posterior avaliação por parte do Pedagogo.

Após concluir a tarefa, clica-se no botão “parar” e é perguntado ao avaliador o estado de humor de cada uma das fotos das crianças. Isso é importante para posterior avaliação pedagógica e, além disso, também se pede a pontuação GAS da tarefa. Por fim, é gerado o relatório HTML contendo os seguintes dados:

- Nome da criança;
- Nome do experimento;
- Nome do robô;
- Tempo de realização da tarefa;
- Quantidade de movimentos;
- Pontuação utilizando o método GAS;
- Seqüência correta pré-programada, caso exista uma. Neste caso também aparece a quantidade de movimentos correto (percentual).

Além dessas informações contidas na parte inicial do relatório há a lista de movimentos feitos pela criança durante a execução das tarefas.

Esta lista contém o número do movimento, sendo que são excluídos os movimentos que fazem com que o robô pare. Por isso apenas o que gera movimento no robô (e não o faça parar) é registrado. Juntamente com o número do movimento existe uma foto da criança que será seguida por uma seta indicando o movimento. Isto é, se a seta é para a direita, o movimento foi para a direita, e assim por diante. O humor da criança no momento da foto é colocado ao lado da mesma, após a foto ser analisada pelo Pedagogo.

Se não houver seqüência correta, isto é, se os movimentos forem livres, as setas aparecem na cor azul. Se houver seqüência correta de movimentos há uma avaliação de acerto do movimento. Em outras palavras, o sistema avalia se o movimento está certo ou errado e pode calcular estatísticas de erro e acertos.

Com isso o relatório apresenta todos os movimentos (sejam certos ou errados), mesmo que os movimentos errados não sejam enviados para o robô. É importante assinalar que o sistema grava os movimentos errados, pois para avaliação pedagógica e estatística estes dados são necessários, embora o robô não se mova com tais movimentos errôneos.

Em termos de exibição no relatório as setas aparecem, então, na cor verde, caso o movimento esteja correto, ou na cor vermelha, caso o movimento esteja errado.

Por fim, o relatório mostra o tempo em que cada movimento foi feito em relação ao primeiro movimento, que aparece como tendo sido feito no instante zero.

Todo esse processo ocorre para cada movimentação registrada pelo sistema durante a experiência e pode ser posteriormente avaliada por um Psicólogo ou Pedagogo. As fotos são de extrema importância para a avaliação do estado emocional das crianças, algo que pode acarretar muitas diferenças nos resultados dos testes.

Ao longo do tempo, tendo vários relatórios das crianças e, automaticamente, um histórico de experiências feitas por elas, é possível um avaliador identificar se houve uma boa adaptação da criança ao sistema, se houve melhorias cognitivo-comportamentais e até mesmo melhorias na parte física.

Os relatórios HTML gerados poderão ser impressos em um browser para arquivamento ou documentação. A figura 22 mostra um relatório gerado pelo programa. A imagem teve que ser reduzida devido ao espaço disponível para a mesma, porém em um browser o relatório aparece nas proporções corretas e de forma nítida.

Nome da Criança: Criança #1
Nome do Teste: Tarefa #1
Nome do Robô: Robô #1

Tempo Gasto: 00:00:12
GAS Score: 2

Movimentação:

- 1  ↑ Movimentação para frente - 00:00:00 :: Humor #1
- 2  ↓ Movimentação para trás - 00:00:08 :: Humor #2
- 3  → Movimentação para direita - 00:00:09 :: Humor #3
- 4  ← Movimentação para esquerda - 00:00:10 :: Humor #4

Figura 23 – Relatório gerado com o programa. As imagens foram borradas propositalmente para não identificar o usuário.

É possível observar a existência de um número antes da fotografia e, após a fotografia tem-se respectivamente, a seta indicando o movimento, um texto informando para onde foi a movimentação, o que complementa a função de ilustração da seta e, em seguida o tempo em que houve o movimento.

Capítulo 5: Tarefas

Os tipos de tarefas a serem realizadas pelas crianças foram sugeridos por uma Pedagoga, a fim de auxiliar na parte educacional e avaliação pedagógica. Além disso, algumas tarefas, já presentes na literatura também foram inseridas – como a tarefa de desenho.

Todas as tarefas têm um acompanhamento de um Pedagogo que avalia de forma mais subjetiva o desempenho da criança quanto à utilização do robô para completar os testes.

As tarefas a serem realizadas com as crianças são a movimentação de blocos lógicos e desenho.

- **Tarefa inicial (treino)** – Mover o robô através de caminhos com obstáculos

Como uma tarefa inicial, as crianças devem mover o robô de tal forma que o mesmo percorra um caminho entre os obstáculos e chegue ao destino desejado. Os obstáculos são algumas almofadas as quais são colocadas no chão a fim de obrigar as crianças a conduzirem o robô da forma correta.

Esta primeira tarefa tem como principal objetivo treinar as crianças de acordo com a movimentação do robô obrigando-as, para concluir a tarefa, a fazer todos os movimentos.

- **Tarefa #1** – Manipulação do robô para que façam um desenho

Esta tarefa consiste em pedir para a criança controlar o robô para fazer um desenho em um papel tamanho A0 que se encontra sob o robô. O controle se dá movimentando o robô nas quatro direções, bem como abaixando o pincel para escrever e levantando-o para movimentar sem desenhar (figura 23).

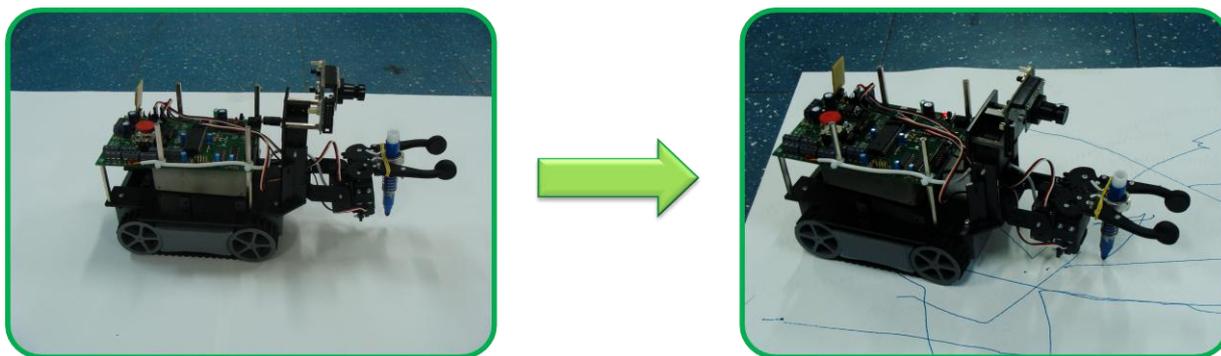


Figura 24 - Desenho - (a) O robô pronto para desenhar; (b) o robô já tendo desenhado alguns traços.

A tarefa de desenho permite a criança desenhar livremente, porém para realizar uma avaliação estatística, são colocados no papel pontos através dos quais a criança deverá fazer um desenho. Em outras palavras, o desenho deve passar por todos esses pontos.

- **Tarefa #2 – Trajetórias com blocos coloridos**

Os blocos coloridos (figura 24) são utilizados como obstáculos e, para desviar dos mesmos e andar pelo caminho pré-estabelecido, a criança deverá fazer todas as movimentações do robô.

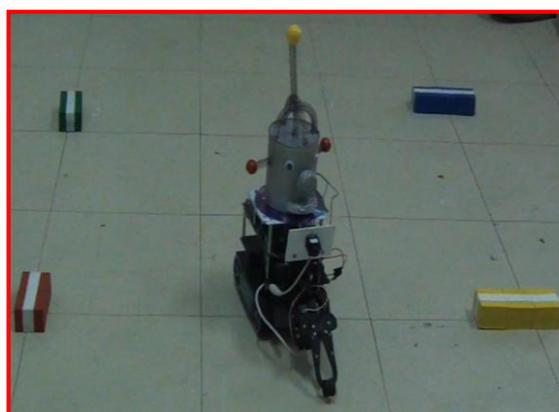


Figura 25 - Tarefa de trajetória com blocos coloridos.

Capítulo 6: Abordagem pedagógica

Para ser possível a avaliação da criança é necessário se ter algumas identificações sobre a criança antes de realizar os testes propostos nos capítulos seguintes desta Dissertação. Para isso, é feita uma série de perguntas aos pais e aos profissionais do CREFES para poder conhecer o estado atual da criança em termos cognitivos e físicos. Além disso, é de grande importância saber quais são as limitações da criança, quais as tarefas que ela gosta de fazer e como a dificuldade motora atrapalha sua vida cotidiana. Tais informações são utilizadas para a avaliação do sistema desenvolvido.

Esta série de perguntas traça o perfil de como a criança lida com a dificuldade física que possui e como isso poderá ser amenizado com o uso da Robótica Educacional. Para quantificar a capacidade da criança em utilizar o robô foi feita a escala (relativa à capacidade motora) apresentada na tabela 1. São feitas comparações da criança antes e depois dos testes, para saber se houve ou não ganho motor (além do ganho cognitivo).

A ambientação do CREFES foi feita utilizando cartazes coloridos com bichos (como se fosse um pequeno sítio) e foram posicionadas as câmeras para filmagem do robô e da criança, além do próprio robô e sensor. Os papéis juntamente com os obstáculos eram colocados e retirados durante os testes conforme necessário. Toda essa ambientação era necessária para dar suporte à parte Pedagógica e ajudar a entreter e interagir com as crianças.

Tabela 1 – Classificação da criança com relação à capacidade motora de utilizar o robô.

Código	Situação
A	Pessoa sem deficiência física
B	Pessoa com deficiência física apenas nos membros inferiores (não interfere na utilização do robô)

C	Pessoa que consegue mexer as mãos e a cabeça, mas possui um atraso na movimentação ou reflexos involuntários leve
D	Pessoa que consegue mexer as mãos e a cabeça, mas possui um atraso na movimentação ou reflexos involuntários médio
E	Pessoa que consegue mexer as mãos e a cabeça, mas possui um atraso na movimentação ou reflexos involuntários grande
F	Pessoa que pode mexer a cabeça, porém não mexe as mãos ou possui grande reflexo involuntário nas mãos
G	Sem controle de movimentação de cabeça e movimentação de membros superiores
H	Poucas condições de utilizar o robô. Controle muito rarefeito da cabeça e nenhum controle ou movimento das mãos.

É importante ter essa avaliação prévia, pois o modo como as crianças irão operar o robô varia consideravelmente de acordo com o tipo de deficiência motora que apresentam. Para a posterior avaliação pedagógica tais informações podem interferir na nota dada no GAS e também na avaliação do comportamento.

6.1 Utilização do método de avaliação GAS

Como definido anteriormente há a necessidade da definição de objetivos claros para a utilização do GAS. Tal definição é subjetiva, porém uma vez feito isso, as notas são dadas de acordo com o sucesso ou não em alcançar o objetivo – o que faz com que esta parte não seja subjetiva.

A tabela 5 mostra os objetivos a serem alcançados para cada tarefa e suas pontuações. A avaliação do GAS será dada pela quantidade de movimentos válidos (a idéia de movimentos válidos ocorre quando o comando dado pela criança ao robô é para se mover e não para parar) com relação à quantidade de movimentos total das crianças que têm condição de utilizar o robô (A-D da tabela 1).

Tabela 2 – Avaliação das tarefas utilizando o GAS.

Avaliação GAS	
Movimentos Válidos (%)	GAS
$N < \text{Controle} * 0,5$	-2
$\text{Controle} * 0,5 < N < \text{Controle} * 0,7$	-1
$\text{Controle} * 0,7 < N < \text{Controle} * 1,05$	0
$\text{Controle} * 1,05 < N < \text{Controle} * 1,1$	1
$T < \text{Controle} * 1,1$	2

Vale a pena ressaltar que N é o número de movimentos válidos e a variável “Controle” é o número médio de movimentos feitos pelas crianças.

Capítulo 7: Resultados

Os experimentos foram realizados em três semanas, sendo que cada criança era avaliada de uma a duas vezes por semana. As que realizam o experimento mais de uma vez tinham como segunda tarefa, uma repetição da primeira ou um desenho livre que não eram computados. Desta forma, a divisão das tarefas foi feita da seguinte maneira:

Na primeira semana a tarefa feita foi a realização de percurso na qual se tinha que movimentar o robô de tal forma que ele conseguisse mover-se em um percurso determinado. Eram utilizados movimentos de cabeça apenas. Essa tarefa inicial serve para ajudar a criança a entender melhor o robô.

Na segunda semana era pedido para as crianças que desenhassem algo no papel passando por pontos marcados.

Na terceira semana era feito um percurso com o robô passando por blocos de acordo com a cor. Serve como uma avaliação se houve um ganho na utilização do robô após as duas primeiras semanas.

É importante também ter conhecimento sobre cada uma das patologias presentes nas crianças. Para proteger a identidade das mesmas seus nomes foram substituídos por numeração de 1 a 14, tal como mostrado na tabela 3.

Tabela 3 – Lista de patologias das crianças e suas habilidades em utilizar o robô.

ID	Patologia	Classificação
1	Paralisia Cerebral	D
2	Distrofia Muscular	C
3	Distrofia Muscular	C
4	Distrofia Muscular	C
5	Atetose, Paralisia Cerebral	H

6	Distrofia Muscular	C
7	Paralisia Cerebral	H
8	Paralisia Cerebral	H
9	Distrofia Muscular	D
10	Paralisia Cerebral	H
11	Diagnóstico ainda não fechado	H
12	Distrofia Muscular	C
13	Distrofia Muscular	C
14	Paraplégica	B

7.1 Resultados observados

Para medir os resultados numericamente foi utilizado o número de movimentos válidos, isto é, aqueles que não resultam em parar o robô. Ao utilizar esta variável é possível mensurar se a criança entendeu bem o funcionamento do robô, pois uma vez mais segura, a mesma tende a controlar melhor a cabeça e parar menos o robô. Por isso a relação de movimentos válidos dividido pela movimentação total tende a ser mais alta.

Um detalhe importante a ressaltar é que apenas as crianças que tem condição de utilizar o robô (com classificação de A até D na tabela de habilidade de utilizar o robô) foram catalogadas no GAS. Além disso, em todas as semanas é utilizado o percentual médio de movimentos válidos como base para o GAS e é feita uma tabela (que estão representadas nas tabelas 4, 6 e 8) utilizando a equação (1) e a tabela 2 com a pontuação GAS.

Posteriormente estas tabelas de pontuação são utilizadas na coluna GAS – MV (onde MV significa movimentos válidos percentuais) nas tabelas de desempenho das crianças (tabelas 5, 7 e 9).

- **Primeira semana:** o teste feito na semana de 01/02/2011 até 03/02/2011 foi de realização de percurso utilizando o robô (tarefa inicial). Os resultados estão representados na tabela 5. A quantidade de movimentos válidos média é de 74,58%

Desta forma, utilizando a equação (1) e a tabela 2, tem-se que a pontuação na primeira semana (na tabela 4):

Tabela 4 – Pontuação GAS da primeira semana.

Avaliação GAS	
Movimentos Válidos – MV (%)	GAS
N<37%	30
37%<N<52%	40
52%<N<78%	50
78%<N<82%	60
N>82%	70

Tabela 5 – Resultados da primeira semana. As porcentagens são relativas à média de movimentos válidos na semana.

Data	ID	Resultado da Tarefa	GAS (MV)
01/02/2011	1	Conseguiu guiar muito bem o robô e completou as tarefas com sucesso. Levou em conta o atraso na comunicação existente no sistema e conseguiu entender a tarefa integralmente.	50 (56,53%)
	4	Teve ótima interação com o robô. Conseguiu desenhar utilizando o sistema e completou o percurso de forma satisfatória. A interação com o robô e o entendimento da tarefa foram muito bons.	70 (92,46%)
	6	Teve o primeiro contato com o robô neste dia. É necessário levar isto em conta, pois quase todas as outras crianças já tiveram contato com o robô anteriormente. Como já tem 7 anos conseguiu entender satisfatoriamente o robô e seus movimentos. Conseguiu fazer os percursos de forma satisfatória.	70 (117,48%)
02/02/2011	7	Entendeu bem a movimentação no sentido para mover para frente o robô. Não conseguiu fazer as tarefas da forma esperada, no entanto é necessário levar em conta que possui apenas 3 anos e possui quadriparesia. Sendo assim possui certa dificuldade de compreensão da tarefa. Queria ficar olhando para o robô ao invés de mexer o boné com a placa.	-
	8	Compreendeu que mexendo o boné conseguia conduzir o robô, especialmente para os lados. No entanto não conseguiu movimentar o robô de forma que conseguisse realizar o percurso. Chorava toda vez que o robô parava ou perdia a visão do mesmo e estava com o humor um pouco cansado no dia de hoje.	-
	9	Controlou de forma muito satisfatória o robô. Conseguiu realizar as tarefas. Apresentou um pouco de dificuldade devido à patologia, pois não conseguia mover a cabeça para frente sem jogar todo o corpo, porém mesmo assim respondeu muito bem às tarefas.	70 (105,21%)

03/02/2011	11	Apresentou certo desinteresse pelo robô em alguns momentos, porém quando conseguiu concentração gostou muito de conduzir no robô. Aparentemente não conseguiu entender de forma correta como o movimento de cabeça pode interferir em como o robô irá se comportar. Para os lados não mexe muito bem.	-
	8	Gostou muito do robô e conseguiu mexer satisfatoriamente no eixo para frente-trás. Teve mais facilidade de mexer com os braços, pois assim não deixava de ver o robô.	-
	12	Primeira vez que tem contato com o robô. Foi apresentado aos movimentos iniciais. Tem 10 anos de idade e teve uma ótima compreensão e controle do robô. Pôde realizar as tarefas com extrema facilidade.	70 (101,14%)

- **Segunda semana:** na primeira semana pode-se ter uma forma de treinamento com o robô. Já na segunda semana foi utilizado o robô para fazer desenhos simples ligando pontos pré-definidos sobre um papel. A tabela 6 apresenta os resultados deste experimento.

Desta forma, utilizando a equação (1) e a tabela 2, têm-se a tabela 6:

Tabela 6 – Pontuação para a segunda semana.

Avaliação GAS Score	
Movimentos Válidos – MV (%)	GAS
N<40%	30
40%<T<56%	40
56%<N<84%	50
84%<N<88%	60
N>88%	70

Tabela 7 – Resultados da segunda semana.

Data	ID	Resultado da Tarefa	GAS (MV)
08/02/2011	2	Conseguiu controlar satisfatoriamente o robô. Completou a tarefa de desenho passando o pincel por todas as marcações brancas. Apresentou muita facilidade no uso do sistema.	50 (74,78%)
	14	Conseguiu entender bem os comandos do robô. Sua patologia restringe a movimentação apenas nos membros inferiores e, desta forma, pôde controlar de forma bem precisa o robô. Teve um pouco de dificuldade na movimentação, mas é necessário levar em conta que foi seu primeiro dia.	60 (85,80%)
	5	Entendeu bem a movimentação do robô. Esta criança não consegue controlar o tônus muscular, sendo que há momentos de relaxamento extremo e momentos de contração extrema. Devido a isso não consegue ter um controle fino do robô, porém a movimentação dos olhos (que são facilmente controlados) mostra que há um entendimento do robô.	-

	6	Controlou o robô extremamente bem. Já na semana passada reconheceu de forma muito rápida os movimentos do robô. É, inclusive, capaz de ajustar a posição da cabeça mesmo se o sensor estiver fora do lugar. Completou a tarefa de forma rápida e satisfatória.	50 (76,35%)
	13	Rapidamente entendeu os movimentos do robô e o conduziu de forma satisfatória e em pouco tempo completou a tarefa de desenho. O resultado foi muito acima do esperado, inclusive, pois não havia tido contato com o robô na semana anterior.	50 (82,49%)
09/02/2011	8	Comandou melhor o robô em relação às vezes anteriores. Fica extremamente alegre ao ver o POB-BOT e muitas vezes não mexe, pois não quer perder a visão do robô. O controle motor da cabeça é rarefeito e não consegue colocar a cabeça na posição correta.	-
	1	Conseguiu mexer muito bem o robô e passar por todas as etiquetas brancas, conforme havia sido combinado na tarefa. Apresenta muita facilidade na condução do robô.	50 (74,78%)
	10	Não gosta de utilizar o boné nem a pulseira na mão. Foi pedido que utilizasse a plaquinha diretamente na mão e, desta forma, mexeu o robô. Pareceu ter mais controle com a placa na mão, porém ficou com um pouco de medo do robô. Não conseguiu fazer a tarefa plenamente.	-
10/02/2011	11	A mãe o auxiliou a utilizar a plaquinha e conseguiu desenhar desta forma. Parece entender corretamente o funcionamento do motor, porém não consegue manter a musculatura firme em uma determinada posição. Quer ficar olhando o robô o tempo todo e, por isso, mexe a cabeça fazendo o robô andar de forma errada.	-
	5	Conseguiu entender o funcionamento do robô de forma satisfatória. Não consegue mexer bem devido a falta de controle muscular.	-
	8	Conseguiu entender bem a movimentação do robô. Mexeu para frente e para trás compreendendo bem a movimentação. Mostrou sinais de entendimento da movimentação lateral.	-
	12	Tem uma ótima coordenação motora apesar da distrofia muscular. Compreendeu rapidamente o objetivo do projeto e conseguiu conduzir muito bem o robô.	50 (73,85%)

- **Terceira semana:** última semana de uso do robô. O principal objetivo da tarefa desta semana foi movimentar o robô entre cubos coloridos. É uma tarefa de realização de percurso como na primeira, porém com objetos menores. Os resultados estão mostrados na tabela 9.

O número de movimentos e a pontuação está na tabela 8.

Tabela 8 – Pontuação para última semana.

Avaliação GAS	
Movimentos Válidos – MV (%)	GAS
N<40%	30
40%<T<56%	40
56%<N<84%	50
84%<N<88%	60
N>88%	70

Tabela 9 – Resultados da última semana.

Data	ID	Resultado da tarefa	GAS (MV)
15/02/2011	3	Primeira vez que tem contato com o robô. Conseguiu manuseá-lo muito bem e de forma muito correta. Conseguiu fazer o teste de passar pelos cubos sem bater. O controle foi acima do esperado já que não tinha visto o robô antes.	50 (73,85%)
	14	Conseguiu interagir muito bem com o robô. Sua compreensão é bastante rápida e pôde mexer o robô muito bem. O desempenho também foi acima do esperado, pois é a segunda vez que teve contato com o robô.	50 (87,01%)
	4	Conseguiu passar pelos cubos coloridos com o robô. Desviou muito bem dos obstáculos e conseguiu resultado satisfatório. Tem muita facilidade em controlar o robô.	50 (80,00%)
	5	Com a ajuda da mãe conseguiu controlar o robô. A criança em si parece ter entendido apenas que o robô vai para frente quando a cabeça é movida nesta direção. Em contrapartida os olhos vão sempre para a posição correta.	-
	6	Conseguiu conduzir no robô muito bem. Tem uma capacidade motora limitada devido a distrofia, no entanto, é capaz de conduzir o robô de forma muito satisfatória. Não teve dificuldades e entendeu completamente a tarefa e os movimentos.	70 (89,69%)
16/02/2011	7	Parece entender bem que o robô se move de acordo com o boné, porém não associou corretamente os movimentos. Aparentemente associou a idéia de que mexendo o boné o robô se move apenas.	-
	8	Mexeu da mesma forma que nas outras semanas. Como tem pouca idade parece ainda não entender muito bem que ele comanda o robô, no entanto parece ter dado sinais de compreensão da movimentação.	-
	9	Conduziu muito bem o robô. Repetiu a experiência 3 vezes, tendo facilidade inclusive de controlar o boné quando o mesmo girou um pouco em sua cabeça (soube compensar o fato do sensor estar girado)	60 (85,41%)
17/02/2011	12	Com um histórico de bom uso do robô, esta criança conseguiu fazer a tarefa com extrema facilidade. Além disso, repetiu a tarefa utilizando um pincel e um papel. Ainda assim conseguiu fazê-la de forma rápida e eficaz.	50 (68,38%)

7.2 Resultados estatísticos por semana

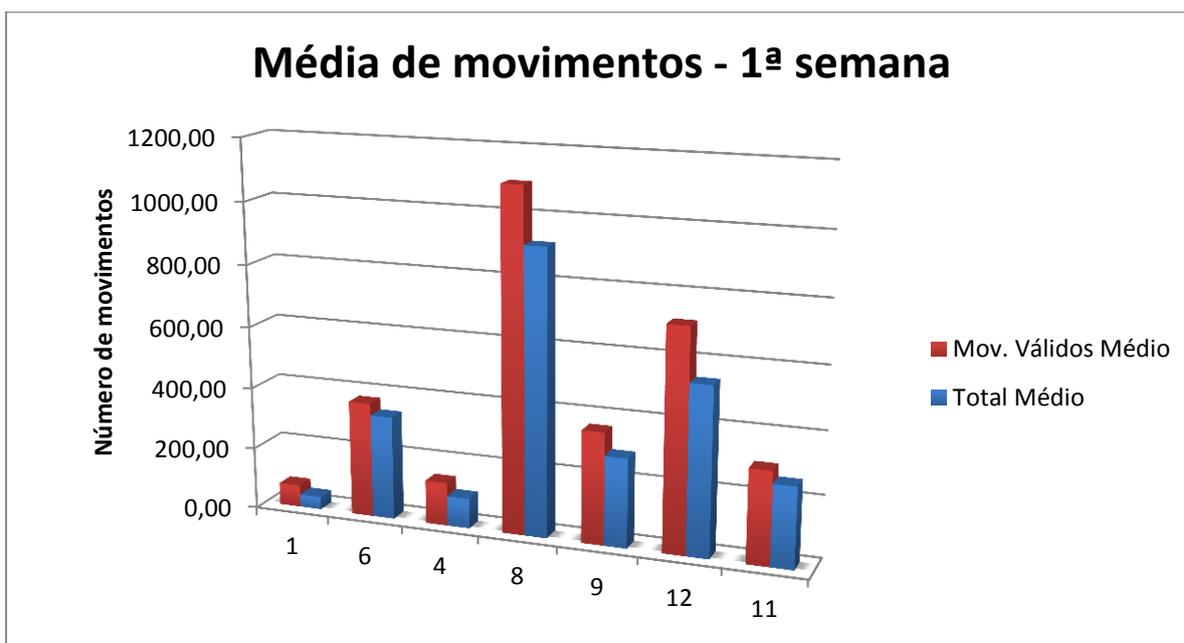
A tabela 10 mostra os tempos e o número de movimentos feitos por cada criança na primeira semana. As colunas desta tabela são:

- Tempo Médio – tempo médio, em minutos, que cada criança demorou para realizar o experimento. Como cada criança realizou mais de uma vez o experimento foi calculado o tempo médio para a realização do experimento;
- Mov. Válidos Médio – é o número de vezes que a criança fez um movimento de cabeça que mandou o robô se locomover. É importante ressaltar que nesta coluna só são registrados os movimentos que fazem o robô mexer e não parar;
- Total Médio – representa o número de movimentos total médio, tendo a mesma idéia da coluna anterior, porém considerando também as movimentações de cabeça da criança que fazem o robô parar;
- Mov./s – exibe a quantidade de movimentos que a criança fez por segundo (em média) em cada tarefa;
- Mov. Válidos – percentual de movimentos de cabeça feitos que mandaram o robô se mexer em comparação com o total de movimentos. Matematicamente é a razão da coluna de movimentos válidos médio dividido pela coluna de movimentos total médio.

Tabela 10 – Resultados estatísticos da primeira semana.

ID	Tempo Médio (min.)	Mov. Válidos Médio	Total Médio	Mov./s	Mov. Válidos
1	01:29	41,16	71,74	0,80	56,53%
6	04:25	335,00	373,00	1,41	89,81%
4	01:27	97,80	140,40	1,47	70,69%
8	06:55	915,67	1095,75	2,49	84,77%
9	02:22	287,64	361,18	2,30	80,43%
12	04:03	541,33	712,67	2,98	75,43%
11	03:11	260,25	300,00	1,32	84,49%
Média	03:42	332,98	409,05	1,68	78,50%

Para uma melhor compreensão, os dados desta tabela estão representados nas figuras 25, 26 e 27.

**Figura 26** - Média de movimentos feitas por cada criança durante a primeira tarefa.

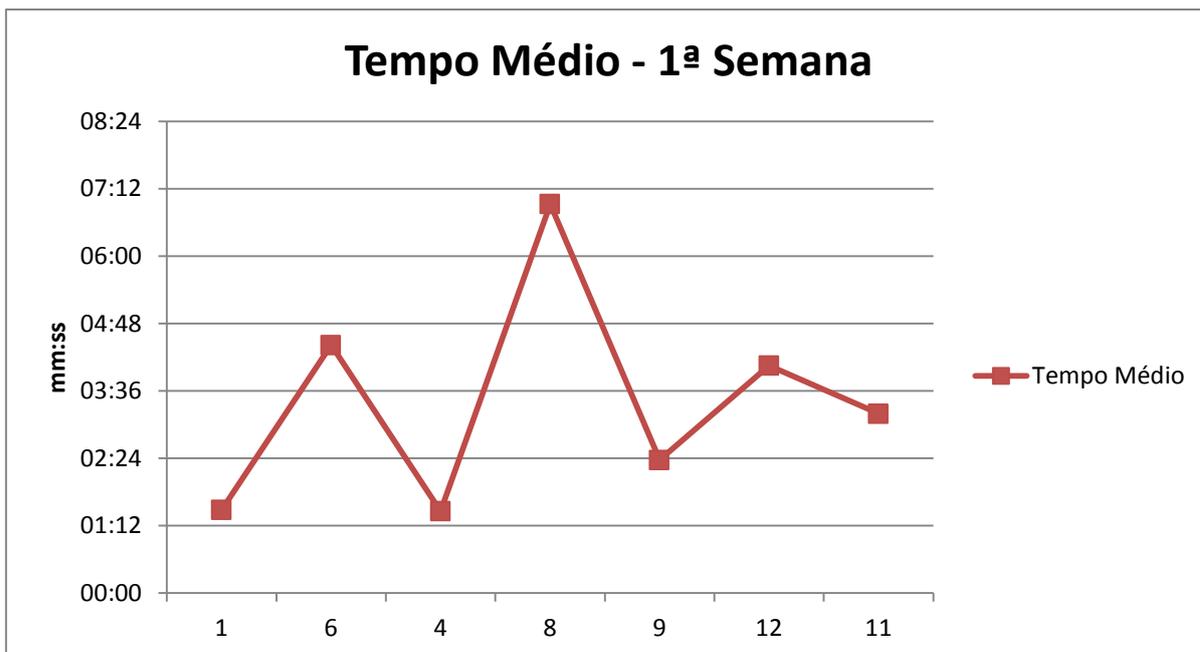


Figura 27 - Tempo médio que cada criança demorou no treino.

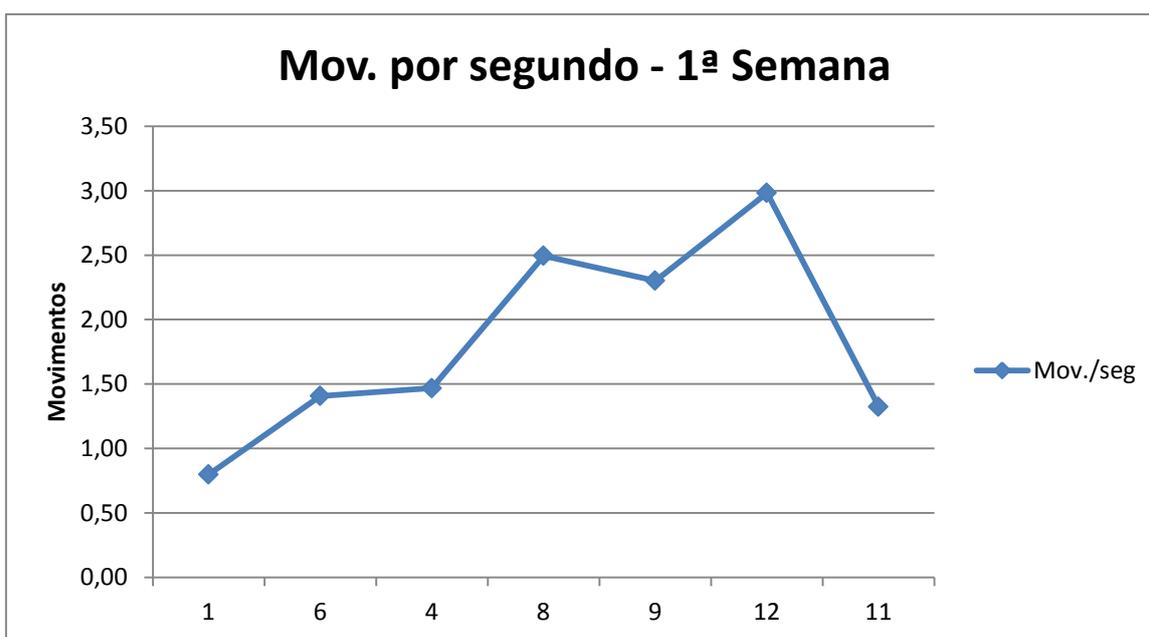


Figura 28 - Quantidade de movimentos feitos por segundo por cada criança.

A tabela 11 mostra os resultados da segunda semana de experimentos, na qual foram realizados os experimentos de desenho com trajetória e desenho livre.

Tabela 11 – Resultados estatísticos da segunda semana.

ID	Tempo Médio (min.)	Mov. Válidos Médio	Total Médio	Mov./s	Mov. Válidos
1	05:20	482,33	683,00	1,86	74,78%
2	02:19	253,67	318,67	2,28	79,30%
5	05:45	509,83	715,67	1,97	72,82%
6	01:32	116,00	143,00	2,32	76,35%
8	05:01	266,75	338,00	0,79	83,20%
9	03:03	239,80	299,00	1,40	89,58%
10	01:54	172,67	217,00	2,56	70,32%
11	03:20	234,50	290,75	1,43	83,19%
12	05:04	314,80	407,40	1,64	73,85%
13	05:24	429,67	483,00	1,49	82,49%
14	02:24	207,40	247,20	1,56	87,01%
Média	03:44	293,40	376,61	1,75	79,35%

Nas figuras 28, 29 e 30 é possível visualizar graficamente as informações da tabela 8.

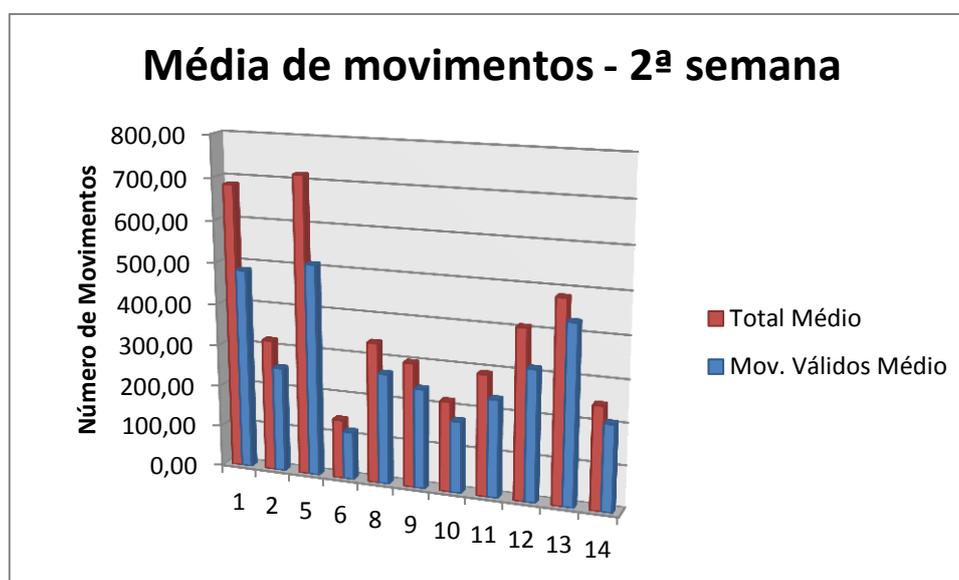
**Figura 29** - Média de movimentos feitos por cada criança na segunda semana.



Figura 30 - Tempo médio para execução da tarefa (por criança) na segunda semana.

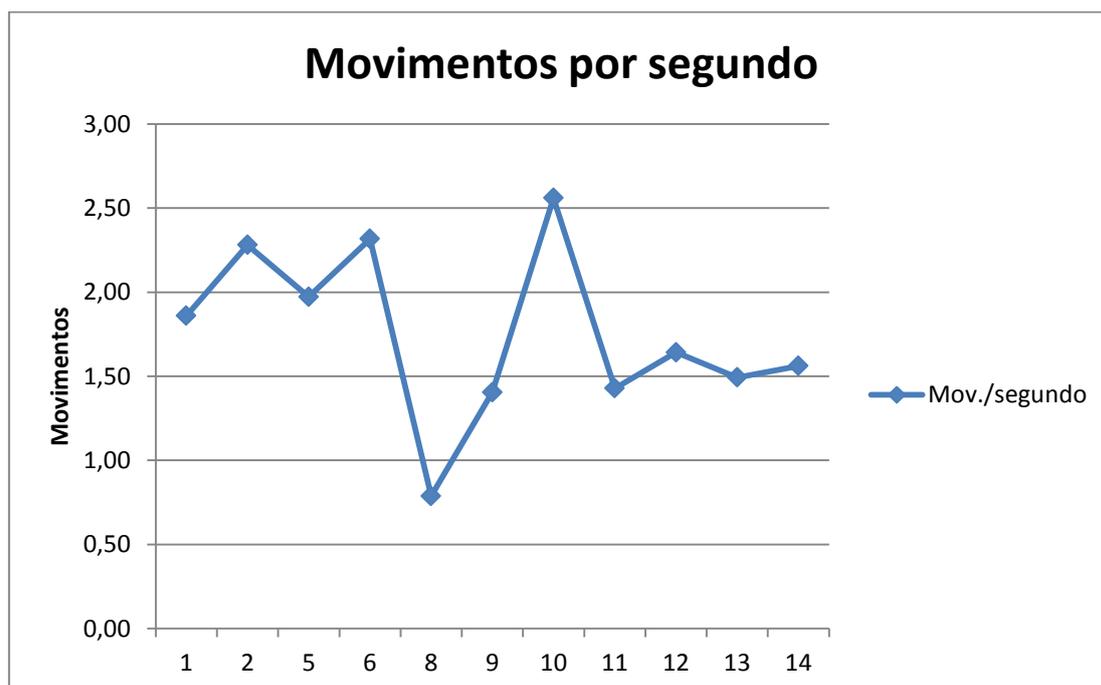
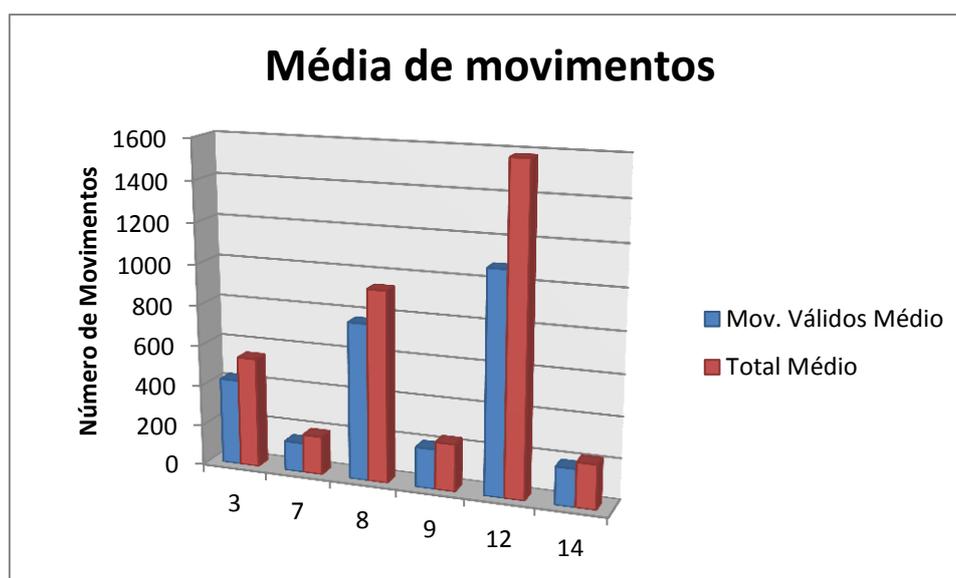


Figura 31 - Número de movimentos médio por segundo na segunda semana.

Por fim, a tabela 12 mostra os resultados da última semana de experimentos. Já as figuras 31, 32 e 33 é a representação gráfica dessa tabela.

Tabela 12 – Resultados estatísticos da terceira semana

ID	Tempo Médio	Mov. Válidos Médio	Total Médio	Mov./s	Mov. Válidos
3	06:16	424	541	1,44	78,37%
7	0,13958	145,66	189	1,354282	71,74%
8	0,45486	768	934	1,403613	78,25%
9	0,18333	196,5	232	0,882239	85,40%
12	13:21	1079	1578	1,97	68,38%
14	05:49	185	217,67	0,63	85,80%
<i>Média</i>	07:21	466,36	615,27	1,28	77,99%

**Figura 32** - Média de movimentações feita na última semana.

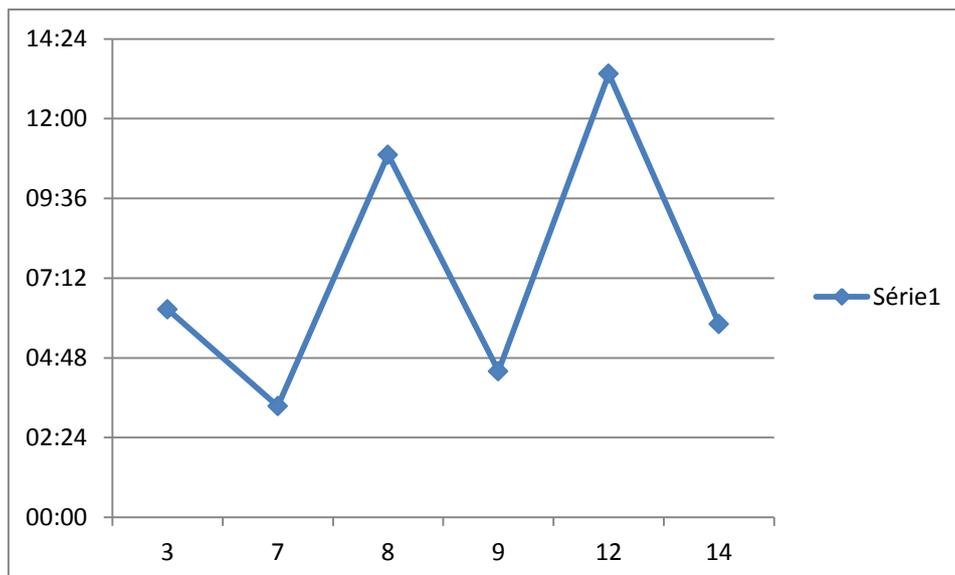


Figura 33 - Tempo médio dos testes na última semana.



Figura 34 - Quantidade média de movimentos por segundo de cada criança na última semana.

7.3 Comparações entre as semanas

É necessário fazer uma comparação entre as semanas para visualizar o uso do robô. A variável que pode ser utilizada para este tipo de comparação é o número de

movimentos a cada segundo, pois a tendência, se a criança souber comandar bem o robô, é que os movimentos fiquem mais precisos e, portanto, o número seja menor.

A tabela 13 mostra os tempos necessários para realizar os experimentos de cada semana.

Tabela 13 – Resultados estatísticos da primeira semana.

ID	Tempo de Movimentação (minutos)		
	Semana 1	Semana 2	Semana 3
1	01:29	05:20	-
2	-	02:19	-
3	-	-	06:16
4	01:27	-	-
6	04:25	01:32	-
9	02:22	03:03	04:24
12	04:03	05:04	13:21
13	-	05:24	-
14	05:49	02:24	-
Média	03:15	03:35	08:00

Na figura 34 é possível observar os tempos para cada tarefa. O tempo do treino e da tarefa de desenho é bem próximo, pois eram mais simples. No entanto, a tarefa de trajetória é mais complicada e, por isso, requer mais tempo.

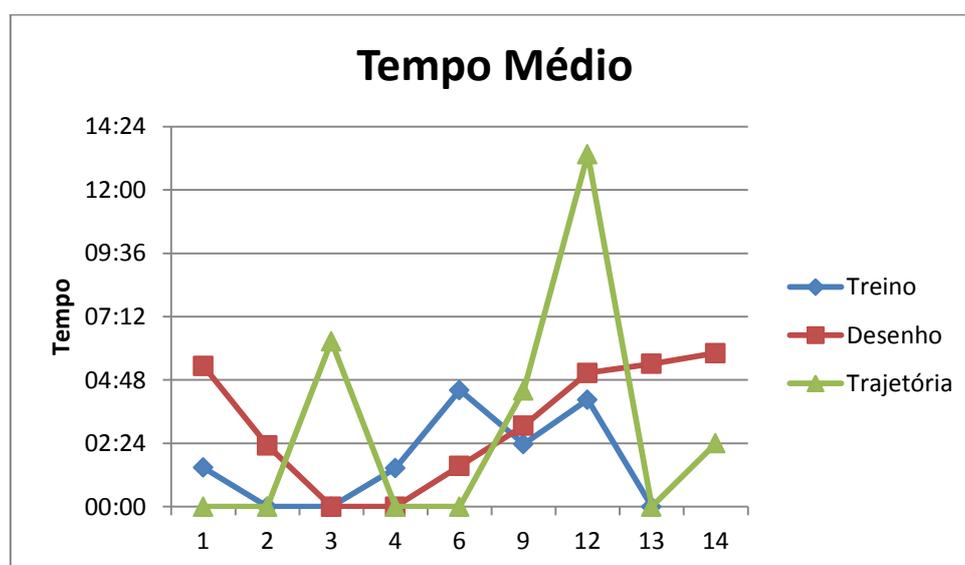


Figura 35 - Tempo médio dos testes na última semana.

Outra variável importante a se medir é a quantidade de movimentação por segundo. Ao longo do tempo, mais seguras sobre como utilizar o robô, as crianças tendem a diminuir o número de movimentos por segundo. Isso pode ser observado na tabela 14.

Tabela 14 – Resultados estatísticos da primeira semana.

Movimentação por segundo			
ID	Semana 1	Semana 2	Semana 3
1	0,80	1,86	-
2	-	2,28	-
3	-	-	1,44
4	1,47	-	-
6	1,41	2,32	-
9	2,30	1,40	0,882239198
12	2,98	1,64	1,97
13	-	1,49	-
14	0,63	1,56	-
Média	1,60	1,79	1,43

Na figura 35 o gráfico exibe a quantidade de movimentos feitos a cada segundo. Observe que, de forma geral, a quantidade de movimentos por segundo cai ao longo do tempo, indicando que as crianças estão mais seguras e sabendo melhor como utilizar o sistema.

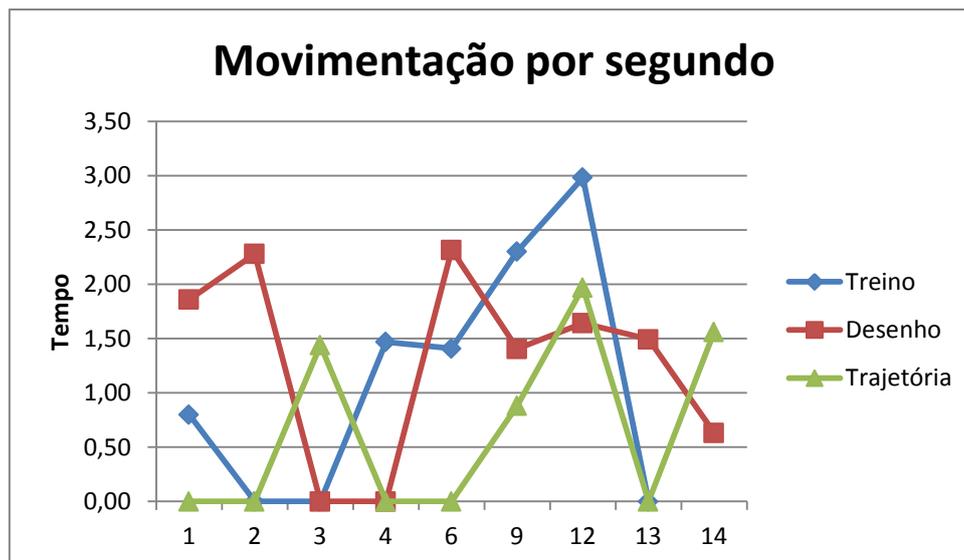


Figura 36 - Tempo médio dos testes na última semana.

A relação entre movimentos válidos e o total de movimentos verifica se a criança entende que pode movimentar o robô trocando de direção e/ou sentido sem parar o mesmo. Logicamente tarefas mais complexas como o desvio de trajetória podem exigir mais paradas do robô, pois os movimentos necessitam ser mais precisos. No entanto, pode-se observar que em comparação com a primeira semana houve um aumento da relação de movimentação e comandos para o robô parar, o que indica que as crianças souberam utilizar melhor o sistema. A quantidade de movimentos válidos nas três semanas pode ser visto na tabela 15.

Tabela 15 – Resultados estatísticos das três semanas.

Relação Movimentos Válidos/Total			
ID	Semana 1	Semana 2	Semana 3
1	56,53%	74,78%	-
2	-	79,30%	-
3	-	-	78,37%
4	70,69%	-	-
6	89,81%	76,35%	-
9	80,43%	89,58%	85,41%
12	75,43%	73,85%	68,38%
13	-	82,49%	-
14	85,80%	87,01%	-
Média	76,45%	80,48%	77,93%

A figura 36 mostra os resultados da tabela 14 em forma gráfica. Observe que é possível ver que, na média, a linha de trajetória e de desenho estão levemente acima da linha de treino, mostrando que a porcentagem de movimentos válidos (isto é, movimentos que não são para o robô parar) cresceu.

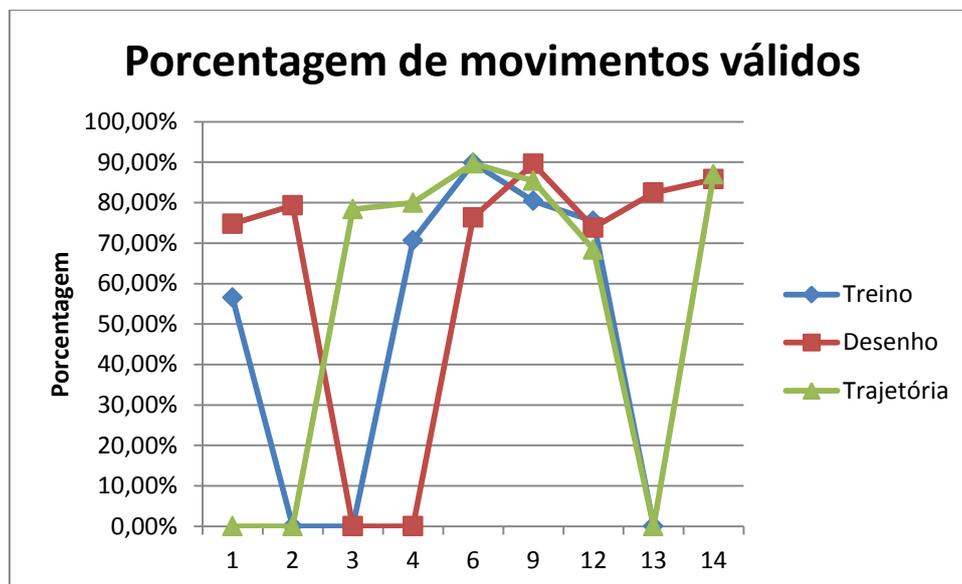


Figura 37 - Média de movimentos válidos por criança nas três semanas.

Capítulo 8: Conclusões

A principal meta deste sistema de Robótica Educacional é fazer com que haja uma melhoria de comunicação entre a criança e o mundo exterior e, por consequência, que esta consiga ter um melhor aprendizado e melhor interação social, fatores importantes no desenvolvimento humano.

De acordo com os resultados obtidos e o relatório de ganho de aprendizagem e desempenho, com o passar do tempo podem ser criadas novas tarefas, que terão a função de aumentar ainda mais a capacidade de interação destas crianças e desenvolver mais a parte cognitiva das mesmas.

A utilização do robô como entretenimento pode ser observada já que as crianças ficaram muito felizes e satisfeitas com a utilização do robô, mesmo aquelas que nem tinham condições de comandá-lo bem. A atenção das crianças ficou muito fixa no robô.

Outro ponto importante a ser considerado é que as crianças eram obrigadas a mexer com a cabeça (e, portanto movimentar-se) para comandar o robô. Com isso algumas acabam por forçar movimentos que não tinham facilidade e, com o tempo, há chances de que melhorem o controle desta movimentação, o que é muito bom para o desenvolvimento motor da criança.

Para a realização deste trabalho, foram realizados os seguintes desenvolvimentos:

- 1) O sensor foi adaptado para ser utilizado para a captura dos sinais biológicos da criança;
- 2) Foi desenvolvida uma interface de computador para registrar os relatórios, contendo as atividades realizadas pelas crianças, além de registrar expressões faciais;
- 3) Foi feita a adaptação do robô comercial POB-BOT para ser possível sua utilização no projeto.

Como conclusão do uso da robótica pelas crianças, pode-se observar que as mesmas tiveram grande gosto pelo robô, bem como souberam utilizá-lo cada vez melhor, sempre interagindo com o mesmo como um brinquedo. Em longo prazo é muito provável que as crianças consigam comandá-lo com grande habilidade, visto que houve grande melhora em apenas três semanas de uso, o que se pode constatar nas tabelas e gráficos apresentados, verificando-se uma maior segurança das crianças e uma melhoria gradativa no uso do sistema desenvolvido, o que é indicativo de uma melhora na sua capacidade cognitiva e também motora.

Além disso, como é uma ferramenta lúdica, pode ajudar, de forma agradável, a criança a fazer algumas tarefas que, por suas condições físicas, não conseguia realizar. Isto certamente irá melhorar o seu desenvolvimento em termos de cognição e socialização.

Finalmente vale ressaltar que as seguintes publicações foram realizadas durante este Projeto de Pesquisa:

- Valadão, Carlos; Bastos Filho, T. F.; Bôrtole, Magdo; Perim, Victor; Celino, Ricardo; Rodor, Fauzi; Gonçalves, Agda; Ferasoli, Humberto; Lópes, Natalia. Educational Robotics as a Learning Aid for Disabled Children. In: ISSNIP Biosignals and Biorobotics Conference 2011, p. 6pg. [23]
- Valadão, Carlos ; Bastos Filho, T. F. ; Bôrtole, Magdo . Uso da Robótica para Exploração e Descoberta do Ambiente para Crianças com Deficiência Física Severa. In: VI Jornadas Argentinas de Robótica, 2010, Buenos Aires. p. 142-147. [24]

Referências Bibliográficas

[1] Cook, Albert M.; Howery, Kathy – Robot-enhanced discovery and exploration for very young children with disabilities – DINF (Disability Information Resources), artigo postado na web em 2 de dezembro de 1998. Site do DINF: www.dinf.ne.jp.

[2] Thomas, RM (1992). *Comparing Theories of Child Development* (3ª ed). Belmont CA: Wadsworth Publishing.

[3] Linder TW (1990). *Transdisciplinary Play-based Assessment; Functional Approach to Working with Young Children*. Baltimore: PH Brooks.

[4] Reportagem sobre Lev Vygotsky na Educar para Crescer - <http://educarparacrescer.abril.com.br/aprendizagem/lev-vygotsky-307440.shtml>

[5] Brinker RP & Lewis M (1982). Discovering the competent handicapped infant: a process approach to assessment and intervention. *Topics in Early Childhood Spec. Educ.*, 2(2), p1-15.

[6] Todia B, Irvin LK, Singer GHS, & Yovanoff P. (1993). The self-esteem parent program. IN Singer GHS & Powers LE (eds) *Families, Disability, and Empowerment*. Toronto: Paul H Brookes.

[7] Seligman, M.E.P. (1975). *Helplessness: On Depression, Development, and Death*. San Francisco: W.H. Freeman. ISBN 0-7167-2328-X

[8] Scherzer AL & Tscharnuter I (1990). *Early Diagnosis and Therapy in Cerebral Palsy*. (2ª ed.) New York: Marcel Dekker.

[9] Swinth Y, Anson D, Deitz J. (1993). Single-switch computer access for infants and toddlers. *American Journal of Occupation Therapy*, 47 (11), 1031-1038.

[10] Cook AM, Hoseit P, Liu KM, Lee RY & Zeteno CM: Using a Robotic Arm System to Facilitate Learning in Very Young Disabled Children, *IEEE Trans Bio. Med. Engr.*, BME-35:132-137, 1998.

[11] Cook, Albert M.; Hoseit. Paul; Liu, Ka Man; Lee, Ronald Y. - Using an arm robotic system to facilitate learning in very young disabled children - IEEE Transactions on biomedical engineering, vol. 35, Nº 2, Fevereiro 1988.

[12] Cook, Albert M.; Meng, M.; Dong, F.; Gu, J. - Using A Robot Arm to Facilitate the Learning in Young and Mentally Development-Delayed Children – ART (Advanced Robotics and Teleoperation) Lab, University of Alberta, Canada.

[13] Cook, Albert M.; Heng Max Q.; Gu, Jason J.; Howery, Kathy - Development of a Robotic Device for Facilitating Learning by Children Who Have Severe Disabilities - IEEE Transactions on biomedical engineering, vol. 10, Nº 3, Setembro 2002.

[14] Werry I.P. and Dautenhahn K. Applying mobile robot technology to the rehabilitation of autistic children. Proc. 7th Int. Symp. Intelligent Robotics Systems (SIRS'99), Coimbra, Portugal, July 1999.

[15] Michaud F. and Clavet A. Robotoy contest—designing mobile robotic toys for autistic children. Proc. American Society for Engineering Education (ASEE'01) Albuquerque, NM, 2001.

[16] Cook A. M., Bentz B., Harbottle N., Lynch C., Miller B. — School-Based Use of a Robotic Arm System by Children With Disabilities. IEEE Trans Neural Systems and Rehabilitation Engineering. Vol. 13, Dec, 2005.

[17] Séguin, P. – POB-BOT User's Manual – POB Technology, Outubro, 2005.

[18] Documento contido no FTP da UFRN sobre modelo cinemático de robô móvel - ftp://users.dca.ufrn.br/sist_rob/cap02.pdf

[19] Bôrtole M. – Projeto de Graduação intitulado “*Desenvolvimento de um sensor inercial para um sistema de avaliação motora*” – UFES, Vitória, Outubro, 2009.

[20] Roving Networks – Datasheet do módulo Bluetooth DS-RN41-V3.1 – Roving Networks, 8 de Abril de 2009

[21] Ribershop - Kit Placa de Captura EasyCap + Kit Câmera Sem Fio - www.ribershop.com.br. Acessado em julho de 2010.

[22] C3 Technology - <http://www.c3technology.com.br> . Acessado em janeiro de 2011.

[23] Valadão C.; Freire Bastos, T., Bôrtole M., Perim V., Celino D. Rodor Fauzi, Gonçalves A., Ferasoli H. Agda Gonçalves, Humberto Ferasoli – *Educational Robotics as a Learning Aid for Disabled Children* – UFES, Vitória, Setembro 2010.

[24] Valadão, Carlos ; Bastos Filho, T. F. ; Bôrtole, Magdo . Uso da Robótica para Exploração e Descoberta do Ambiente para Crianças com Deficiência Física Severa. In: VI Jornadas Argentinas de Robótica, 2010, Buenos Aires. p. 142-147.

[25] José Carlos Macoratti – VB.NET – www.macoratti.net . Acessado em julho de 2010.

[26] MSDN (Microsoft Developer Network) – www.msdn.com.

Apêndice I: Programação feita em Visual Basic

A programação da interface no computador foi feita em Visual Basic Express Edition 2008 que é grátis e possui suporte ao .NET Framework. Essa interface compõe-se de quatro telas de cadastro, uma de relatórios e a janela principal de testes. Antes disso, aparece um menu com as opções do programa – a janela *início*. Para fazer o programa foi utilizado como apoio o site VB.NET – José Macoratti, [25] e MSDN (Microsoft Developer Network), [26].

É importante citar que o código aqui descrito foi o utilizado para realizar os testes, portanto está configurado apenas para o robô padrão e o sensor inercial (padrão). Embora o banco de dados já esteja preparado para guardar os comandos que vem do sensor e os comandos que devem ser enviados para o robô, o código utilizado não faz a conversão relacionando os bancos de dados, mas sim diretamente no código.

Outra importante questão é que neste código a calibração foi feita manualmente inserindo valores dentro do código. O código do FastCalibration (que será mostrado mais a frente) pode alterar estes valores.

Além deste código há outros trechos de código auxiliares, alguns gerados automaticamente pelas configurações no Visual Basic. Para simplificar, neste apêndice estão apenas as partes principais do programa.

- **Módulo Global #1 – Variáveis Globais**

O módulo global #1 guarda as variáveis globais utilizadas em várias partes do programa, sendo que uma alteração feita em uma parte irá afetar a execução de outra parte que também utilize a mesma variável. Isso serve para fazer uma integração entre as diversas partes do programa.

- **Variáveis relativas ao banco de dados**

```
'Variáveis de conexão (1) e salvamento (2) no Banco de Dados
Public ConexaoBD As New SqlServerCe.SqlCeConnection("Data Source=" &
Application.StartupPath & "\GeralDatabase.sdf") '(1)
Public FlagSalvar As Boolean '(2)
```

- **Variáveis para controle de direção e movimento**

```
'Variáveis de controle de direção e movimento
Public direcao As String
Public PosicaoAtual As String
Public ParaMovimentos As Boolean = False
Public FlagSeq As Boolean = False
```

- **Variáveis para controle de tempo**

```
'Variáveis de controle de tempo
'(1) Guarda o momento em que se iniciou a tarefa.
'(2) Guarda o momento em que terminou a tarefa
'(3) Guarda o tempo corrido da tarefa
Public StartTime As TimeSpan
Public StopTime As TimeSpan
Public TempoCorrido As TimeSpan
```

- **Variáveis de controle**

```
'Variáveis de controle
'(1) Indica o status do robô (start/stop)
'(2) Ativa a seqüência correta
'(3) Testa se a FlagSequenciaChecked está ativada
'(4) Guarda o movimento a ser enviado pelo robô
'(5) Permite a execução do programa que gera o relatório
'(6) Mostra se o robô e sensor estão conectados
'(7) Permite iniciar a tarefa
'(8) Pára a tarefa
Public FlagStatus As String = "Start" '(1)
Public FlagSequenciaChecked As Boolean '(2)
Public FlagTestaSequenciaChecked As Boolean '(3)
Public FlagDirecao As String '(4)
Public FlagGeraHTML As Boolean '(5)
Public ConnectionFlag As Integer '(6)
```

```
Public FlagIniciar As Boolean = False      '(7)
Public FlagParar As Boolean = False      '(8)
```

- **Variáveis para relatórios**

```
'Relatórios
' (1) Mostra quantos registros há dentro do banco de dados.
' (2) Guarda o tipo de robô que está sendo utilizado.
' (3) Indica qual nome está selecionado para gerar o relatório.
' (4) Indica qual teste está selecionado para gerar o relatório.
' (5) Indica qual robô está selecionado para gerar o relatório.
' (6) Guarda o relatório em formato texto para gerar o HTML.
' (7) Variável utilizada para gravar se o movimento foi correto ou errado
(C/E)
' (8) Array que guarda até o tempo em que ocorreu uma movimentação
(limitado em 500)
' (9) Serve para montagem dos movimentos corretos.
' (10) Guarda a movimentação correta em formato string (character).
' (11) Guarda o nome do robô utilizado
' (12) Guarda o diretório de destino dos relatórios
' (13) Guarda o número de erros durante a movimentação com sequência
' (14) Array que guarda o humor em cada foto
' (15) Variável para guardar a nota dada ao fim da tarefa
' (16) Calcula o número de movimentos feitos pela criança
' (17) Guarda a movimentação que houve no formato string (caracteres)
Public NumRegistros As Integer           '(1)
Public Tipo_Robo As String                '(2)
Public NomeSelecionado As String         '(3)
Public TesteSelecionado As String        '(4)
Public RoboSelecionado As String         '(5)
Public texto As String                    '(6)
Public VerificaMovimento As String       '(7)
Public TimeAtMovement(100000) As TimeSpan '(8)
Public InitialFlag(100) As Boolean        '(9)
Public Sequencia As String = ""          '(10)
Public marca As String                    '(11)
Public directory As String                '(12)
Public Erro As Integer                    '(13)
Public Humor(0) As String                 '(14)
Public GAS_Score As Integer               '(15)
Public NumMovimentos As Integer           '(16)
Public Movimentacao As String = Nothing  '(17)
```

- **Variáveis de direção**

```
'Variáveis de direção
Public IrEsq As Integer      'Esquerda
Public IrDir As Integer     'Direita
Public IrFrente As Integer  'Frente
Public IrTras As Integer    'Tras
```

- **Variáveis para exibição de valores na tela**

```
'Variáveis para exibição na tela de valores
' getAcelerometro - exibe informação da inclinação nos eixos
' getBateria - exibe informação do nível de bateria
' valores - guarda a string enviada pelo sensor
' SEMG_Medio - guarda sinal médio do SEMG
Public getAcelerometro0 As Integer
Public getAcelerometro1 As Integer
Public getAcelerometro2 As Integer
Public getBateria As Integer
Public valores As String
Public SEMG_Medio As Integer
```

- **Variáveis para animação na tela**

```
'Variáveis para animação na tela
Public OnlyAnimation As Boolean
```

- **Conversão sensor-robô**

```
'Conversão sensor-robô
Public DirRightText As String
Public DirLeftText As String
Public DirUpText As String
Public DirDownText As String
Public ButtonAText As String
Public ButtonBText As String
Public ButtonCText As String
Public ButtonXText As String
Public ButtonYText As String
Public TipoEquipamento As String
```

- **Lista de figuras**

```
'Lista de figuras
Public UpArrow As String = Application.StartupPath &
"\Imagens\OnUpArrow.png"

Public DownArrow As String = Application.StartupPath &
"\Imagens\OnDownArrow.png"

Public LeftArrow As String = Application.StartupPath &
"\Imagens\OnLeftArrow.png"

Public RightArrow As String = Application.StartupPath &
"\Imagens\OnRightArrow.png"

Public PictureNumber As Integer
```

- **Escrita no Bluetooth**

```
'Flag que permite a escrita de sequência de movimentos
Public StartWritingSequence As Boolean
```

- **Dispositivos Auxiliares**

```
'Dispositivos Auxiliares
'Caracteres enviados pelo sensor
Public auxdev_right As Char
Public auxdev_left As Char
Public auxdev_up As Char
Public auxdev_down As Char
Public auxdev_bta As Char
Public auxdev_btb As Char
Public auxdev_btc As Char
Public auxdev_btx As Char
Public auxdev_bty As Char
```

- **Robôs**

```
'Robôs
'Caracteres entendidos pelo robô
Public robot_right As Char
Public robot_left As Char
```

```

Public robot_up As Char
Public robot_down As Char
Public robot_bta As Char
Public robot_btb As Char
Public robot_btc As Char
Public robot_btx As Char
Public robot_bty As Char

```

- **Calibração do sensor inercial**

```

'Calibração do sensor inercial (via software)
Public GlobalIrFrente As Integer = 1900
Public GlobalIrTras As Integer = 2100
Public GlobalIrDireita As Integer = 1900
Public GlobalIrEsquerda As Integer = 2100

```

- **Calibração manual**

```

'Calibração manual
Public IrFrente
Public IrTras
Public IrDir
Public IrEsq

```

- **Módulo Global #2 – Funções Globais**

Este módulo foi construído para abrigar funções que são utilizadas na janela de tarefas. As mesmas não ficaram no arquivo das tarefas, pois o código ficaria muito extenso em uma única tela.

```

Imports System.Runtime.InteropServices

Module Module6

    'Trecho de código retirado e adaptado do site
    ' http://www.macoratti.net/ [25]

    'A primeira parte consiste no carregamento dos parâmetros necessários para
    a DLL

```

```

Public WM_CAP As Short = &H400S

Public WM_CAP_DRIVER_CONNECT As Integer = WM_CAP + 10
Public WM_CAP_DRIVER_DISCONNECT As Integer = WM_CAP + 11
Public WM_CAP_EDIT_COPY As Integer = WM_CAP + 30

Public WM_CAP_SET_PREVIEW As Integer = WM_CAP + 50
Public WM_CAP_SET_PREVIEWRATE As Integer = WM_CAP + 52
Public WM_CAP_SET_SCALE As Integer = WM_CAP + 53
Public WS_CHILD As Integer = &H40000000
Public WS_VISIBLE As Integer = &H10000000
Public SWP_NOMOVE As Short = &H2S
Public SWP_NOSIZE As Short = 1
Public SWP_NOZORDER As Short = &H4S
Public HWND_BOTTOM As Short = 1

Public iDevice As Integer = 0 ' ID do dispositivo atual
Public hWnd As Integer ' manipulador da janela do visualizador
Public ArrayFotos As New ArrayList

'A segunda parte consiste em algumas funções necessárias para manipular a
janela da imagem
Declare Function SendMessage Lib "user32" Alias "SendMessageA" _
(ByVal hWnd As Integer, ByVal wParam As Integer, ByVal lParam As Integer,
<MarshalAs(UnmanagedType.AsAny)> ByVal lParam As Object) As Integer

Declare Function SetWindowPos Lib "user32" Alias "SetWindowPos" (ByVal
hWnd As Integer, _
ByVal hWndInsertAfter As Integer, ByVal x As Integer, ByVal y As Integer,
ByVal cx As Integer, ByVal cy As Integer, ByVal wFlags As Integer) As Integer

Declare Function DestroyWindow Lib "user32" (ByVal hWnd As Integer) As
Boolean

Declare Function capCreateCaptureWindowA Lib "avicap32.dll" (ByVal
lpszWindowName As String, ByVal dwStyle As Integer, _
ByVal x As Integer, ByVal y As Integer, ByVal nWidth As Integer, ByVal
nHeight As Short, ByVal hWndParent As Integer, ByVal nID As Integer) As
Integer

Declare Function capGetDriverDescriptionA Lib "avicap32.dll" (ByVal
wDriver As Short, _
ByVal lpszName As String, ByVal cbName As Integer, ByVal lpszVer As
String, ByVal cbVer As Integer) As Boolean

```

```

'Este módulo é o responsável pela câmera
Public imagem As Image
Public Figura(100000) As Image

'Esta primeira função carrega todas as câmeras conectadas ao computador e
as exibe em um
    'ComboBox no formato de lista.
Public Sub carregaCameras()
    Dim Nome As String = Space(100)
    Dim Ver As String = Space(100)
    Dim bRetorna As Boolean
    Dim x As Integer = 0
    Do
        bRetorna = capGetDriverDescriptionA(x, Nome, 100, Ver, 100)
        If bRetorna Then Tarefas.ComboBox2.Items.Add(Nome.Trim)
        x += 1
    Loop Until bRetorna = False
    Tarefas.PictureBox6.SizeMode = PictureBoxSizeMode.StretchImage
End Sub

'Esta função aponta para a janela (figura) a qual deve ser exibida a
imagem da câmera.
Public Sub abreJanelaVisualizacao(ByVal Picture As Object)
    Dim iHeight As Integer = Tarefas.PictureBox6.Height
    Dim iWidth As Integer = Tarefas.PictureBox6.Width

    iDevice = Tarefas.ComboBox2.SelectedIndex
    'Esta função de sistema captura a imagem e a coloca na janela
    hHwnd = capCreateCaptureWindowA(iDevice, WS_VISIBLE Or WS_CHILD, 0, 0,
40, 30, Tarefas.PictureBox6.Handle.ToInt32, 0)
    SendMessage(hHwnd, WM_CAP_DRIVER_CONNECT, iDevice, 0)

    If SendMessage(hHwnd, WM_CAP_DRIVER_CONNECT, iDevice, 0) Then
        SendMessage(hHwnd, WM_CAP_SET_SCALE, True, 0)
        SendMessage(hHwnd, WM_CAP_SET_PREVIEWRATE, 30, 0)
        SendMessage(hHwnd, WM_CAP_SET_PREVIEW, True, 0)
        SetWindowPos(hHwnd, HWND_BOTTOM, 0, 0, Picture.Width,
Picture.Height, SWP_NOMOVE Or SWP_NOZORDER)
    Else
        DestroyWindow(hHwnd)
    End If
End Sub

'Esta função pára a exibição da imagem da câmera
Public Sub fechaJanelaVisualizacao()
    SendMessage(hHwnd, WM_CAP_DRIVER_DISCONNECT, iDevice, 0)

```

```

        DestroyWindow(hHwnd)
    End Sub

    'Esta função é responsável por salvar uma imagem no formato Bitmap
    Public Sub Salvar(ByVal Picture As Object, ByVal Filename As String)
        Dim data As IDataObject
        Dim bmap As Bitmap

        'Captura imagem da câmera como Bitmap
        SendMessage(hHwnd, WM_CAP_EDIT_COPY, 0, 0)
        data = Clipboard.GetDataObject()
        bmap = CType(data.GetData(GetType(System.Drawing.Bitmap)), Bitmap)

        Dim bmap2 As New Bitmap(80, 60)

        'Grava imagens como bitmap
        Dim g As Graphics = Graphics.FromImage(bmap2)

        'Gera as imagens no relatório apenas se houve algum movimento
        If NumMovimentos > 0 Then

            g.InterpolationMode =
Drawing2D.InterpolationMode.HighQualityBicubic
            g.DrawImage(bmap, New Rectangle(0, 0, 80, 60), New Rectangle(0, 0,
bmap.Width, _
            bmap.Height), GraphicsUnit.Pixel)
            g.Dispose()

            bmap2.Save(directory & "\" & Filename & ".bmp",
Imaging.ImageFormat.Bmp)
        End If
    End Sub

    Public Sub ResetaVariaveis()
        Movimentacao = Nothing
        Erro = 0
        NumMovimentos = 0
        PosicaoAtual = 0
        ParaMovimentos = False
        VerificaMovimento = Nothing
        Tarefas.PictureBox2.Left = 218
        Tarefas.PictureBox2.Top = 145
        FlagGeraHTML = False
    End Sub

```

```

Public Sub GeraHTML()
    MsgBox(My.Computer.Clock.ToString)
End Sub

End Module

```

- **Tela inicial – O Menu**

Esta parte do programa é a tela de boas vindas que mostra as diversas telas que podem ser chamadas dentro do programa.

```

Public Class Inicio

    'Inicia a tela de início e limpa os textos
    Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        TextoCentral.Text = Nothing
        TextoDesc.Text = Nothing
        TextoTitulo.Text = Nothing

        'Tenta conectar ao banco de dados. Caso não consiga exibe erro.
        'O código abaixo será iniciado junto com o programa e refere-se a
conexão com o banco de dados principal chamado aqui de "Geral Database", sendo
ele um SQL Compact Edition.

        Try
            ConexaoBD.Open()
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    'As partes seguintes são responsáveis por fazer a animação das figuras e
textos nesta tela.
    Private Sub PictureBox1_MouseEnter(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles PictureBox1.MouseEnter
        TextoCentral.Text = "Tarefas"
        TextoCentral.TextAlign = ContentAlignment.MiddleCenter
        PictureBox1.Image = Image.FromFile(Application.StartupPath &
"\Imagens\LightBlue.png")
        TextoTitulo.Text = TextoCentral.Text
    End Sub

```

```

        PictureBox7.Image = Image.FromFile(Application.StartupPath &
"\Imagens\BarraAzul.png")
        TextoDesc.Text = "Esta é a parte principal do programa. Contém as
tarefas a serem realizadas e os robôs" _
            & " que podem realizá-las. Também exibe o painel de
controle e permite escolher a tarefa" _
            & " a qual será feita. Para entrar basta clicar no círculo
azul."
    End Sub

    Private Sub PictureBox3_MouseEnter(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles PictureBox3.MouseEnter
        TextoCentral.Text = "Relatório"
        TextoCentral.TextAlign = ContentAlignment.MiddleCenter
        PictureBox3.Image = Image.FromFile(Application.StartupPath &
"\Imagens\LightRed.png")
        TextoTitulo.Text = TextoCentral.Text
        PictureBox7.Image = Image.FromFile(Application.StartupPath &
"\Imagens\RedBar.png")
        TextoDesc.Text = "Aqui pode-se ver os relatórios já gerados pelos
testes feitos anteriormente. Permite " _
            & "a opção de ver em HTML. Para entrar basta clicar no
círculo vermelho."

    End Sub

    Private Sub PictureBox2_MouseEnter(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles PictureBox2.MouseEnter
        TextoCentral.Text = "Testes"
        TextoCentral.TextAlign = ContentAlignment.MiddleCenter
        PictureBox2.Image = Image.FromFile(Application.StartupPath &
"\Imagens\LightYellow.png")
        TextoTitulo.Text = TextoCentral.Text
        PictureBox7.Image = Image.FromFile(Application.StartupPath &
"\Imagens\BarraAmarela.png")
        TextoDesc.Text = "Aqui pode-se montar novos testes, bem como descrever
seus objetivos. Pede-se dados " _
            & "como nome e objetivodo teste. Para entrar basta clicar
no círculo amarelo."

    End Sub

    Private Sub PictureBox5_MouseEnter(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles PictureBox5.MouseEnter
        TextoCentral.Text = "Robôs"
        TextoCentral.TextAlign = ContentAlignment.MiddleCenter

```

```

        PictureBox5.Image = Image.FromFile(Application.StartupPath &
"\Imagens\LightPurple.png")
        TextoTitulo.Text = TextoCentral.Text
        PictureBox7.Image = Image.FromFile(Application.StartupPath &
"\Imagens\BarraRoxa.png")
        TextoDesc.Text = "Nesta seção pode-se fazer o cadastro de robôs.
Define-se seus controles e suas " _
                & "portas de comunicação. Para entrar clique no círculo
roxo."

    End Sub

    Private Sub PictureBox4_MouseEnter(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles PictureBox4.MouseEnter
        TextoCentral.Text = "Cadastro"
        TextoCentral.TextAlign = ContentAlignment.MiddleCenter
        PictureBox4.Image = Image.FromFile(Application.StartupPath &
"\Imagens\LightGreen.png")
        TextoTitulo.Text = TextoCentral.Text
        PictureBox7.Image = Image.FromFile(Application.StartupPath &
"\Imagens\BarraVerde.png")
        TextoDesc.Text = "Nesta seção pode-se fazer o cadastro de pessoas. São
cadastrados dados básicos, " _
                & "como nome, dados dos pais, causa da necessidade
especial, nascimento e foto."

    End Sub

    Private Sub PictureBox6_MouseEnter(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles PictureBox6.MouseEnter
        TextoCentral.Text = "Auxiliares"
        PictureBox6.Image = Image.FromFile(Application.StartupPath &
"\Imagens\LightOrange.png")
        TextoTitulo.Text = TextoCentral.Text
        PictureBox7.Image = Image.FromFile(Application.StartupPath &
"\Imagens\BarraLaranja.png")
        TextoDesc.Text = "Nesta seção define-se os controladores auxiliares,
como acelerômetros, SEMG, EEG. " _
                & "São definidas as portas de comunicação e nome, dentre
outros parâmetros importantes."

    End Sub

    'As rotinas seguintes tratam o clique no botão.
    Private Sub PictureBox2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox2.Click
        AddTarefa.ShowDialog()

```

```

End Sub

Private Sub PictureBox3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox3.Click
    HTML.ShowDialog()
End Sub

Private Sub PictureBox4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox4.Click
    Cadastro.ShowDialog()
End Sub

Private Sub PictureBox5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox5.Click
    Robos.ShowDialog()
End Sub

Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox1.Click
    Tarefas.ShowDialog()
End Sub

Private Sub PictureBox6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox6.Click
    Auxiliares.ShowDialog()
End Sub

Private Sub LinkLabel1_LinkClicked_1(ByVal sender As System.Object, ByVal
e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel1.LinkClicked
    AboutBox2.ShowDialog()
End Sub
End Class

```

- **Janela principal – Tarefas**

Nesta janela é possível configurar todos os parâmetros do experimento (robô, dispositivo, tarefa e identificação da criança), bem com as portas de comunicação, câmera, seqüência correta e calibração. O botão iniciar e parar tarefa, além de um console para substituir os sensores virtualmente, também estão presentes nesta janela.

```

Imports System.Threading

Public Class Tarefas

    'A função seguinte serve para controlar a movimentação do robô e fazer
    as animações.
    Function DirRobo(ByVal direcao As String, ByVal text As String)

        Dim deltaHor As Integer
        Dim deltaVer As Integer

        If OnlyAnimation = False Then
            If FlagIniciar = True Then
                If Bluetooth.IsOpen Then
                    If FlagSequenciaChecked Then
                        If PosicaoAtual < Sequencia.Length Then
                            If Sequencia.ToArray(PosicaoAtual) = direcao Then
                                PosicaoAtual = PosicaoAtual + 1
                                Mov.Text = "Movimento Correto"
                                VerificaMovimento = VerificaMovimento & "C"
                                NumMovimentos = NumMovimentos + 1
                                ColorBox.Image =
Image.FromFile(Application.StartupPath & "\Imagens\LightGreen.png")
                                TimeAtMovement(NumMovimentos) = Now.TimeOfDay
- StartTime

                                Bluetooth.Write(direcao)
                                Label4.Text = "Movimentos: "
                                Label100.Text = NumMovimentos.ToString
                                Salvar(PictureBox6, (NumMovimentos).ToString)
                                Figura(NumMovimentos) = PictureBox6.Image
                            Else
                                Mov.Text = "Movimento Errado"
                                VerificaMovimento = VerificaMovimento & "E"
                                ColorBox.Image =
Image.FromFile(Application.StartupPath & "\Imagens\LightRed.png")
                                Erro = Erro + 1
                                NumMovimentos = NumMovimentos + 1
                                TimeAtMovement(NumMovimentos) = Now.TimeOfDay
- StartTime

                                Label4.Text = "Movimentos: "
                                Label100.Text = NumMovimentos.ToString
                                Salvar(PictureBox6, (NumMovimentos).ToString)
                                Figura(NumMovimentos) = PictureBox6.Image
                            End If '5
                        Else '4
                            MsgBox("Não há mais movimentos a serem feitos.")
                    End If
                End If
            End If
        End If
    End Function
End Class

```

```

        ParaMovimentos = True
        Robo.Write(direcao)
    End If '4
Else '3
    FlagSeq = False
    ColorBox.Image =
Image.FromFile(Application.StartupPath & "\Imagens\LightBlue.png")
    Mov.Text = "Seqüência Livre"
    NumMovimentos = NumMovimentos + 1
    TimeAtMovement(NumMovimentos) = Now.TimeOfDay -
StartTime

    Label4.Text = "Movimentos: "
    Label100.Text = NumMovimentos.ToString
    Robo.Write(direcao)
    Salvar(PictureBox6, (NumMovimentos).ToString)
    Figura(NumMovimentos) = PictureBox6.Image
End If '3
Else '2
    If FlagSequenciaChecked Then
        FlagSeq = "true"
        If PosicaoAtual < Sequencia.Length Then
            If (Sequencia.ToArray(PosicaoAtual)) = direcao
Then
                PosicaoAtual = PosicaoAtual + 1
                Mov.Text = "Movimento Correto"
                VerificaMovimento = VerificaMovimento & "C"
                ColorBox.Image =
Image.FromFile(Application.StartupPath & "\Imagens\LightGreen.png")
                NumMovimentos = NumMovimentos + 1
                TimeAtMovement(NumMovimentos) = Now.TimeOfDay
- StartTime

                Label4.Text = "Movimentos: "
                Label100.Text = NumMovimentos.ToString
                Salvar(PictureBox6, (NumMovimentos).ToString)
                Figura(NumMovimentos) = PictureBox6.Image
            Else
                Mov.Text = "Movimento Errado"
                VerificaMovimento = VerificaMovimento & "E"
                ColorBox.Image =
Image.FromFile(Application.StartupPath & "\Imagens\LightRed.png")
                Erro = Erro + 1
                NumMovimentos = NumMovimentos + 1
                TimeAtMovement(NumMovimentos) = Now.TimeOfDay
- StartTime

                Label4.Text = "Movimentos: "
                Label100.Text = NumMovimentos.ToString

```

```

        Salvar(PictureBox6, (NumMovimentos).ToString)
        Figura(NumMovimentos) = PictureBox6.Image
    End If
Else
    MsgBox("Não há mais movimentos a serem feitos.")
    ParaMovimentos = True
End If
Else
    FlagSeq = False
    ColorBox.Image =
Image.FromFile(Application.StartupPath & "\Imagens\LightBlue.png")
    Mov.Text = "Seqüência Livre"
    NumMovimentos = NumMovimentos + 1
    TimeAtMovement(NumMovimentos) = Now.TimeOfDay -
StartTime

    Label4.Text = "Movimentos: "
    Label100.Text = NumMovimentos.ToString
    Salvar(PictureBox6, (NumMovimentos).ToString)
    Figura(NumMovimentos) = PictureBox6.Image
    If Robo.IsOpen = True Then
        Robo.Write(direcao)
    End If
End If
End If '2
Else
    MsgBox("Clique em Iniciar (círculo verde) para começar a
tarefa")

End If '1
If ParaMovimentos = False Then
    Movimentacao = Movimentacao & direcao
End If
End If '0

Select Case direcao
    Case "d"
        deltaHor = +5
        deltaVer = 0

    Case "a"
        deltaHor = -5
        deltaVer = 0

    Case "s"
        deltaHor = 0
        deltaVer = +5

```

```

        Case "w"
            deltaHor = 0
            deltaVer = -5
        End Select

    Return Nothing

End Function

Public Sub Tarefas_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    carregaCameras()
    StartStop.Image = Image.FromFile(Application.StartupPath &
"\Imagens\StartCircle.png")
    CheckBox1.Checked = True
    CheckBox1.Enabled = False
    DispAux.Items.Add("Padrão")
    DispAux.SelectedItem = "Padrão"
    DispAux.Enabled = False
    RoboBox.Items.Add("Padrão")
    RoboBox.SelectedItem = "Padrão"
    RoboBox.Enabled = False
    ResetaVariaveis()
End Sub

'Os PictureBoxes seguintes são responsáveis pelos botões na guia
controle, inicio e término da tarefa.

Private Sub PictureBox5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles StartStop.Click

    If NomeBox.SelectedItem = Nothing Then
        MsgBox("Selecione o nome de uma pessoa")
    Else
        If TesteBox.SelectedItem = Nothing Then
            MsgBox("Especifique uma tarefa")
        Else
            If (RoboBox.SelectedItem = Nothing) And (CheckBox1.Checked =
False) Then
                MsgBox("Selecione um robô")
            Else
                If FlagStatus = "Start" Then

                    NomeBox.Select()
                    TesteBox.Select()
                End If
            End If
        End If
    End If
End Sub

```

```

        DispAux.Select()
        RoboBox.Select()
        ComboBox2.Select()
        ResetaVariaveis()
        abreJanelaVisualizacao(PictureBox6)
        FlagStatus = "Stop"
        directory = "H:\Testes\" & NomeSelecioneado &
DateString & Now.Hour & Now.Minute & Now.Second
        My.Computer.FileSystem.CreateDirectory(directory)
        StartStop.Image =
Image.FromFile(Application.StartupPath & "\Imagens\StopCircle.png")
        FlagIniciar = True
        FlagParar = False
        ResetaVariaveis()
        LinkLabel3.Hide()
        StartTime = Now.TimeOfDay

    Else
        If Robo.IsOpen = False Then
            Robo.Open()
        End If
        Robo.Write("t")
        Robo.Close()

        fechaJanelaVisualizacao()
        FlagStatus = "Start"
        StartStop.Image =
Image.FromFile(Application.StartupPath & "\Imagens\StartCircle.png")
        FlagIniciar = False
        FlagParar = True
        StopTime = Now.TimeOfDay

        TempoCorrido = StopTime - StartTime

        If NumMovimentos > 0 Then
            NomeSelecioneado = NomeBox.SelectedItem.ToString
            TesteSelecioneado = TesteBox.SelectedItem.ToString
            RoboSelecioneado = RoboBox.SelectedItem.ToString
            FlagGeraHTML = True
            HumorCrianca.ShowDialog()
            MetodoGAS.ShowDialog()
            HTML.ShowDialog()
        Else
            MsgBox("Não ocorreu registros de movimentos nesta
tarefa")
        End If
    End If

```

```

        Sequencia = Nothing
        PosicaoAtual = Nothing
        FlagSequenciaChecked = False
        LinkLabel3.Show()
    End If
End If
End If
End Sub

'Trechos responsáveis por tratar cliques nos botões.
Private Sub AdicionarTeste_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AdicionarTeste.Click
    AddTarefa.ShowDialog()
End Sub

Private Sub AdicionarRobo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AdicionarRobo.Click
    Robos.ShowDialog()
End Sub

Private Sub AdicionarNome_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AdicionarNome.Click
    Cadastro.ShowDialog()
End Sub

Private Sub PictureBox8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BotaoX.Click
    DirRobo("p", "Pinça")
End Sub

Private Sub PictureBox9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BotaoY.Click
    DirRobo("t", "Trás Esq.")
End Sub

Private Sub PictureBox10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BotaoA.Click
    DirRobo("c", "Trás Esq.")
End Sub

Private Sub Button7_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Sair.Click
    'Fecha a janela do programa. Encerra o programa
    Me.Close()
End Sub

```

```

Private Sub PictureBox17_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    If ConnectionFlag = 0 Then
        MsgBox("Verifique a conexão remota e a porta. Caso queira usar
simulação virtual desconsidere esta mensagem.")
    Else
        ConnectionFlag = 1
    End If
End Sub

Private Sub PictureBox18_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    If FlagSequenciaChecked = False Then
        MsgBox("Com esta opção marcada é necessária a configuração correta
dos movimentos. Utilize a tela auxiliar para configurá-los.")
        Sequence.ShowDialog()
    Else
        MsgBox("Não é possível modificar com a tarefa em andamento")
    End If

End Sub

Private Sub LinkLabel3_LinkClicked(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel3.LinkClicked
    Sequence.ShowDialog()
End Sub

Public Sub BluetoothThread()

    Dim UltMov As String = "0"
    'Define variáveis que serão usadas para armazenar os
'dados dos sensores
    Dim BTStr As String
    Dim GetString As String = Nothing
    Dim SEMG(20) As Integer
    Dim Acelerometro(3) As Integer
    Dim Bateria As Integer
    Dim MediaEMG As Integer
    Dim a As Integer = 1

    'Define limite para que haja a movimentação para
'cada lado, baseado no valor recebido pelos sensores.
'Tais valores podem ser alterados dentro da parte de calibração.

```

```

IrFrente = GlobalIrFrente
IrTras = GlobalIrTras
IrDir = GlobalIrDireita
IrEsq = GlobalIrEsquerda

' Caso a porta Bluetooth esteja aberta, a mesma é fechada
' para evitar problemas de comunicação.
If Bluetooth.IsOpen = True Then
    Bluetooth.Close()
End If

Bluetooth.Open()
' Loop da função para que sempre haja leitura do sensor
While True

    If a < 2 And a > 0 Then
        Thread.Sleep(2000)
        a = a - 1
    End If

    ' Abre a porta Bluetooth para iniciar leitura
    ' Bluetooth.Open()

    ' Reseta variável BTStr (Bluetooth String)
    BTStr = ""

    ' Verifica se a leitura do caractere Bluetooth é igual a
    ' letra "i" que indica início do pacote de dados. Se for
    ' começa a "encher" a string BTStr até completar 93 caracteres.
    If (ChrW(Math.Abs(Bluetooth.ReadChar()))) = "i" Then
        For j As Integer = 0 To 93
            BTStr = BTStr & ChrW(Bluetooth.ReadChar())
        Next
        If BTStr(93) <> "i" Then
            BTStr = ""
        End If
    End If

    ' Confirma, para evitar erros, que a string que guarda o
    ' pacote enviado do sensor tem 93 caracteres.
    If BTStr.Length = 94 Then

        ' É feita a separação do sinal em:
        ' (1) 20 amostras de SEMG, cada uma contendo 4 bytes
        ' (2) 3 amostras de acelerômetro, cada uma contendo 4 bytes
        ' (3) 1 byte para mostrar bateria

```

```

For i As Integer = 0 To 19 '1 to 20
    SEMG(i) = Val(BTStr.Substring(i * 4, 4))
    MediaEMG = MediaEMG + SEMG(i)
Next

MediaEMG = MediaEMG / 20

'Como o SEMG_Médio de um sinal equivale ao zero é necessário
fazer um RMS
For i As Integer = 0 To 19 '1 to 20
    SEMG_Medio = SEMG_Medio + Math.Pow((SEMG(i) - MediaEMG),
2) 'Faz (x0-x_avg)^2 + (x1-x_avg)^2 + ...
Next

SEMG_Medio = Math.Pow(SEMG_Medio / 20, 0.5) 'Faz
sqrt(SEMG_Medio/20)

For i As Integer = 20 To 22 '20 to 22
    Acelerometro(i - 20) = Val(BTStr.Substring(i * 4, 4))
Next

Bateria = Val(BTStr(92))

'As variáveis globais a seguir são utilizadas para exibir os
valores

'dos sensores na thread principal.
getAcelerometro0 = Acelerometro(0)
getAcelerometro1 = Acelerometro(1)
getAcelerometro2 = Acelerometro(2) 'getAcelerometro2 =
Acelerometro(2)
getBateria = Bateria

'Caso a tarefa já tenha iniciado o FlagStatus fica em "Stop"
'e quando isso ocorre os comandos podem ser enviados para o
'robô.
If FlagStatus = "Stop" Then

    'As variáveis a seguir servem para executar os comandos
    'na thread principal.
    Dim DRobo As New DirRoboDelegate(AddressOf DirFcn)
    Dim ExibeAcelerometros As New ExibeAcDelegate(AddressOf
ExibeFcn)

    'Comando que exibe na tela os valores dos sensores

```

```

        Me.Invoke(ExibeAcelerometros, getAcelerometro0,
getAcelerometro1, getAcelerometro2, getBateria)

        'Verifica se o robô está conectado. Se não estiver tenta
conectá-lo.

        If Robo.IsOpen = False Then
            Robo.Open()
        End If

        'O acelerômetro 1 é responsável por ir para frente e para
trás.

        'Ele tem prioridade sobre o acelerometro 0 que é
responsável por girar
        'o robô em torno do próprio eixo para esquerda ou para a
direita

        'Funciona da seguinte forma os limites de valor para o
robô se movimentar:
        ' (1) Valores abaixo de IrFrente faz com que o robô se
movimente para frente
        ' (2) Valores acima de IrTras faz com que o robô se
movimente para trás
        ' (3) Valores abaixo de IrDir faz com que o robô vá para
direita
        ' (4) Valores acima de IrEsq faz com que o robô vá para
esquerda
        ' (5) Se nenhum dos IFs anteriores se aplicar o robô
para.

        If Acelerometro(1) < IrFrente Then 'Verifica se o
acelerometro aponta para frente
            If UltMov <> "w" Then
                OnlyAnimation = False
                Me.Invoke(DRobo, "w")
                UltMov = "w"
            Else
                OnlyAnimation = True
                Me.Invoke(DRobo, "w")
            End If
        Else
            If Acelerometro(1) > IrTras Then 'Verifica se o
acelerometro aponta para frente
                If UltMov <> "s" Then
                    OnlyAnimation = False
                    Me.Invoke(DRobo, "s")
                    UltMov = "s"
                End If
            End If
        End If
    End Sub

```

```

Else
    OnlyAnimation = True
    Me.Invoke(DRobo, "s")
End If
Else 'If - Else - If
    If Acelerometro(0) < IrDir Then 'Verifica se o
acelerometro aponta para frente
        If UltMov <> "d" Then
            OnlyAnimation = False
            Me.Invoke(DRobo, "d")
            UltMov = "d"
        Else
            OnlyAnimation = True
            Me.Invoke(DRobo, "d")
        End If
    Else
        If Acelerometro(0) > IrEsq Then 'Verifica se o
acelerometro aponta para frente
            If UltMov <> "a" Then
                OnlyAnimation = False
                Me.Invoke(DRobo, "a")
                UltMov = "a"
            Else
                OnlyAnimation = True
                Me.Invoke(DRobo, "a")
            End If
        Else
            If SEMG_Medio > 100000 Then
                If UltMov <> "g" Then
                    OnlyAnimation = False
                    Me.Invoke(DRobo, "t")
                    Me.Invoke(DRobo, "g")
                    UltMov = "p"
                Else
                    OnlyAnimation = True
                    Me.Invoke(DRobo, "p")
                End If
            Else
                If UltMov <> "t" Then
                    OnlyAnimation = False
                    Me.Invoke(DRobo, "t")
                    UltMov = "t"
                Else
                    OnlyAnimation = True
                    Me.Invoke(DRobo, "t")
                End If
            End If
        End If
    End If
End If

```

```

                End If
            End If 'Robô ir para esquerda
        End If
    End If 'Robô ir para direita
End If 'Robô ir para trás
End If 'Robô ir para frente
End If 'Responsável por verificar se já
começou a tarefa (FlagStatus)
End If 'Responsável pela verificação do
tamanho da string

End While 'Fim do While

End Sub

'Os Delegates servem para intermediar as threadings. Isto é, são
utilizadas
'para ser possível a realização de multi-threading.
Private Delegate Sub DirRoboDelegate(ByVal direcao As String)
Private Delegate Sub LabelDelegate(ByVal str As String)
Private Delegate Sub EscreveRoboDelegate(ByVal direcao As String)
Private Delegate Sub EscreveAngulo(ByVal Acelerometro As Integer)
Private Delegate Sub ExibeAcDelegate(ByVal getAcelerometro0 As Integer,
ByVal getAcelerometro1 As Integer, ByVal getAcelerometro2 As Integer, ByVal
getBateria As Integer)
Private Delegate Sub GetStringDelegate(ByVal GetString As String)
Private Delegate Sub TextBoxDelegate(ByVal BTStr As String)

'Função que envia o comando para a função DirRobo na thread principal.
Private Sub DirFcn(ByVal direcao As String)
    DirRobo(direcao, "")
End Sub

'Rotina para conectar o sensor.
Private Sub LinkLabel2_LinkClicked(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
SensorLabel.LinkClicked

    Try
        Thread.Sleep(1000)
        Bluetooth.PortName = SensorCOM.SelectedItem.ToString
        Bluetooth.Open()
        ConexaoBT.Image = Image.FromFile(Application.StartupPath &
"\Imagens\LightGreen.png")

        If Robo.IsOpen Then
            ConnectionFlag = True

```

```

        End If

        'Criação das nova thread e inicialização
        Dim BTThread As New Threading.Thread(AddressOf BluetoothThread)
        BTThread.Start()

        'Caso dê errado o erro é capturado pela variável 'ex' que guarda
as propriedades do erro
        Catch ex As Exception
            MsgBox(ex.Message)
            ConnectionFlag = False
            ConexaoBT.Image = Image.FromFile(Application.StartupPath &
"\Imagens\LightRed.png")
        End Try

    End Sub

    'Rotina para conectar o robô.
    Private Sub LinkLabel4_LinkClicked(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
RoboLabel.LinkClicked

        Try
            Thread.Sleep(1000)
            Robo.PortName = RoboCOM.SelectedItem.ToString
            Robo.Open()
            ConexaoRobo.Image = Image.FromFile(Application.StartupPath &
"\Imagens\LightGreen.png")

            If Bluetooth.IsOpen Then
                ConnectionFlag = True
            End If

            'Caso dê errado o erro é capturado pela variável 'ex' que guarda
as propriedades do erro
            Catch ex As Exception
                MsgBox(ex.Message)
                ConnectionFlag = False
                ConexaoRobo.Image = Image.FromFile(Application.StartupPath &
"\Imagens\LightRed.png")
            End Try

        End Sub

        Private Sub ComboBox1_MouseClick(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventHandler) Handles SensorCOM.MouseClick

```

```

SensorCOM.Items.Clear()
Dim aux As String
For Each STR As String In My.Computer.Ports.SerialPortNames
    If STR.Length > 4 Then
        aux = STR.Substring(0, 4)
        STR = aux
    End If
    SensorCOM.Items.Add(STR)
Next
End Sub

Private Sub ComboBox5_MouseClick(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles RoboCOM.MouseClick
    RoboCOM.Items.Clear()
    Dim aux As String
    For Each STR As String In My.Computer.Ports.SerialPortNames
        If STR.Length > 4 Then
            aux = STR.Substring(0, 4)
            STR = aux
        End If
        RoboCOM.Items.Add(STR)
    Next
End Sub

Private Sub AdicionarDisp_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles AdicionarDisp.Click
    Auxiliares.ShowDialog()
End Sub

Private Sub NomeBox_MouseClick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles NomeBox.MouseClick
    'Inicia variáveis para leitura do banco de dados
    Dim strSql As String
    Dim strdr As SqlServerCe.SqlCeDataReader

    'Envia o comando para o banco de dados ler tudo do Tarefas
    'e jogar para a variável strdr_tarefa
    strSql = "SELECT * FROM Cadastro"
    Dim cmd As New SqlServerCe.SqlCeCommand(strSql, ConexaoBD)
    strdr = cmd.ExecuteReader()

    'Preenche a ComboBox com a leitura da strdr_tarefa, sendo que
    'primeiramente limpa qualquer dado escrito antes na ComboBox para
    'evitar duplicatas.
    NomeBox.Items.Clear()
    While (strdr.Read())

```

```

        NomeBox.Items.Add(strrdr("Nome").ToString)
    End While
End Sub

Private Sub ComboBox4_MouseClick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TesteBox.MouseClick
    'Inicia variáveis para leitura do banco de dados
    Dim strSql_tarefa As String
    Dim strrdr_tarefa As SqlServerCe.SqlCeDataReader

    'Envia o comando para o banco de dados ler tudo do Tarefas
    'e jogar para a variável strrdr_tarefa
    strSql_tarefa = "SELECT * FROM Tarefas"
    Dim cmd As New SqlServerCe.SqlCeCommand(strSql_tarefa, ConexaoBD)
    strrdr_tarefa = cmd.ExecuteReader()

    'Preenche a ComboBox com a leitura da strrdr_tarefa, sendo que
    'primeiramente limpa qualquer dado escrito antes na ComboBox para
    'evitar duplicatas.
    TesteBox.Items.Clear()
    While (strrdr_tarefa.Read())
        TesteBox.Items.Add(strrdr_tarefa("Nome").ToString)
    End While
End Sub

Private Sub RoboBox_MouseClick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RoboBox.MouseClick
    'Inicia variáveis para leitura do banco de dados
    Dim strSql_Robos As String
    Dim strrdr_Robos As SqlServerCe.SqlCeDataReader

    'Envia o comando para o banco de dados ler tudo do Robos
    'e jogar para a variável strrdr_tarefa
    strSql_Robos = "SELECT * FROM Robos"
    Dim cmd As New SqlServerCe.SqlCeCommand(strSql_Robos, ConexaoBD)
    strrdr_Robos = cmd.ExecuteReader()

    'Preenche a ComboBox com a leitura da strrdr_tarefa, sendo que
    'primeiramente limpa qualquer dado escrito antes na ComboBox para
    'evitar duplicatas.
    RoboBox.Items.Clear()
    While (strrdr_Robos.Read())
        RoboBox.Items.Add(strrdr_Robos("Nome").ToString)
    End While

```

```

End Sub

Private Sub DispAux_MouseClick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DispAux.MouseClick
    'Inicia variáveis para leitura do banco de dados
    Dim strSql_Auxiliares As String
    Dim strdrdr_Auxiliares As SqlServerCe.SqlCeDataReader

    'Envia o comando para o banco de dados ler tudo do Robos
    'e jogar para a variável strdrdr_tarefa
    strSql_Auxiliares = "SELECT * FROM Dpositivos_Auxiliares"
    Dim cmd As New SqlServerCe.SqlCeCommand(strSql_Auxiliares, ConexaoBD)
    strdrdr_Auxiliares = cmd.ExecuteReader()

    'Preenche a ComboBox com a leitura da strdrdr_tarefa, sendo que
    'primeiramente limpa qualquer dado escrito antes na ComboBox para
    'evitar duplicatas.
    DispAux.Items.Clear()
    While (strdrdr_Auxiliares.Read())
        DispAux.Items.Add(strdrdr_Auxiliares("Equipamento").ToString)
    End While

End Sub

'Rotinas para movimentar o robô quando se clica nas setas da interface.
Private Sub Frente_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Frente.Click
    DirRobo("w", "Frente")
End Sub

Private Sub Esquerda_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Esquerda.Click
    DirRobo("a", "Esquerda")
End Sub

Private Sub Tras_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Tras.Click
    DirRobo("s", "Trás")
End Sub

Private Sub Direita_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Direita.Click
    DirRobo("d", "Direita")
End Sub

```

```

Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.LinkLabelLinkClickedEventArgs)

    'Variáveis responsáveis pelos movimentos (associação no banco de
dados)

    Dim DispFrente As String
    Dim DispTras As String
    Dim DispDireita As String
    Dim DispEsquerda As String
    Dim DispBotaoX As String
    Dim DispBotaoY As String
    Dim DispBotaoA As String
    Dim DispBotaoB As String
    Dim DispBotaoC As String

    'Variáveis responsáveis pela aquisição dos sinais (associação no banco
de dados)

    Dim RoboFrente As String
    Dim RoboTras As String
    Dim RoboDireita As String
    Dim RoboEsquerda As String
    Dim RoboBotaoX As String
    Dim RoboBotaoY As String
    Dim RoboBotaoA As String
    Dim RoboBotaoB As String
    Dim RoboBotaoC As String

    'Leitura no banco de dados

    '(1) Variáveis para leitura do banco de dados
    Dim sql_rdr_disp As SqlServerCe.SqlCeDataReader
    Dim sql_rdr_robo As SqlServerCe.SqlCeDataReader

    '(2) Comandos a serem enviados para o SQL
    Dim sql_disp As String = "SELECT * FROM Dispositivos_Auxiliares WHERE
(Equipamento = '" & DispAux.SelectedItem & "')"
    Dim sql_robo As String = "SELECT * FROM Robos WHERE (Nome = '" &
RoboBox.SelectedItem & "')"

    '(3) Variáveis para rodar os comandos
    Dim sql_cmd_disp As New SqlServerCe.SqlCeCommand(sql_disp, ConexaoBD)
    Dim sql_cmd_robo As New SqlServerCe.SqlCeCommand(sql_robo, ConexaoBD)

    '(4) Jogando os dados para as variáveis de leitura
    sql_rdr_disp = sql_cmd_disp.ExecuteReader

```

```

sql_rdr_robo = sql_cmd_robo.ExecuteReader

'(5) Associando os dados nas strings de recepção
DispFrente = sql_rdr_disp("DirUp").ToString
DispTras = sql_rdr_disp("DirDown").ToString
DispDireita = sql_rdr_disp("DirRight").ToString
DispEsquerda = sql_rdr_disp("DirLeft").ToString
DispBotaoA = sql_rdr_disp("BotaoA").ToString
DispBotaoB = sql_rdr_disp("BotaoB").ToString
DispBotaoC = sql_rdr_disp("BotaoC").ToString
DispBotaoX = sql_rdr_disp("BotaoX").ToString
DispBotaoY = sql_rdr_disp("BotaoY").ToString

'(6) Associando os dados nas strings de envio
RoboFrente = sql_rdr_robo("DirUp").ToString
RoboTras = sql_rdr_robo("DirDown").ToString
RoboDireita = sql_rdr_robo("DirRight").ToString
RoboEsquerda = sql_rdr_robo("DirLeft").ToString
RoboBotaoA = sql_rdr_robo("BotaoA").ToString
RoboBotaoB = sql_rdr_robo("BotaoB").ToString
RoboBotaoC = sql_rdr_robo("BotaoC").ToString
RoboBotaoX = sql_rdr_robo("BotaoX").ToString
RoboBotaoY = sql_rdr_robo("BotaoY").ToString

'(7) Exibindo as strings para verificar
Dim Texto As String = ""
Texto = Texto & DispFrente & " => " & RoboFrente & vbNewLine
Texto = Texto & DispTras & " => " & RoboTras & vbNewLine
Texto = Texto & DispDireita & " => " & RoboDireita & vbNewLine
Texto = Texto & DispEsquerda & " => " & RoboEsquerda & vbNewLine
Texto = Texto & DispBotaoA & " => " & RoboBotaoA & vbNewLine
Texto = Texto & DispBotaoB & " => " & RoboBotaoB & vbNewLine
Texto = Texto & DispBotaoC & " => " & RoboBotaoC & vbNewLine
Texto = Texto & DispBotaoX & " => " & RoboBotaoX & vbNewLine
Texto = Texto & DispBotaoY & " => " & RoboBotaoY & vbNewLine

MsgBox(Texto)
End Sub

Private Sub LinkLabel1_LinkClicked_1(ByVal sender As System.Object, ByVal
e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel1.LinkClicked
    FastCalibration.ShowDialog()
End Sub

End Class

```

- **Cadastro de crianças**

Esta parte do programa é responsável por guardar no banco de dados os nomes das crianças. É possível colocar, além do nome da criança, os nomes dos pais e a deficiência física inerente à criança.

```
Imports System.Runtime.InteropServices

Public Class Cadastro

    'Carrega itens de banco de dados
    Private Sub CadastroBindingNavigatorSaveItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
        'Me.Validate()
        'Me.CadastroBindingSource.EndEdit()
        'Me.TableAdapterManager1.UpdateAll(Me.DatabaseCadastroDataSet)
    End Sub

    Private Sub Cadastro_Load_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the
'DatabaseCadastroDataSet.Cadastro' table. You can move, or remove it, as
needed.
        'Me.CadastroTableAdapter.Fill(Me.DatabaseCadastroDataSet.Cadastro)

        'Carrega os cadastros já feitos para possíveis alterações ou exclusões
RetornaForm()

        TotalRegistros()
        NumIDTextBox.Enabled = False
        NumIDTextBox.Text = NumRegistros + 1

    End Sub

    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
        Me.Close()
    End Sub

    Private Sub RetornaForm()

        'Esta rotina serve para carregar o número de tarefas já existentes e
limpar os campos de nome e descrição.
```

```

        'Todo o procedimento de conexão com banco de dados e envio de comando
e string são feitos.
        Dim sql As String = "SELECT * FROM Cadastro"
        Dim comm As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)
        Dim streader As SqlServerCe.SqlCeDataReader

        Try
            streader = comm.ExecuteReader
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

        'Limpa alguns campos
        NomeTextBox1.Clear()
        CausaNecEspTextBox.Clear()
        NomeMaeTextBox.Clear()
        NomePaiTextBox.Clear()
        ComboBox1.Items.Clear()

        'Atualização das tarefas para pesquisa
        While streader.Read()
            ComboBox1.Items.Add(streader("Nome"))
        End While
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click

        'Esta parte do código impede que haja dois cadastros com o mesmo nome
ou número
        Dim FlagBlockSaving As Boolean = False

        'Declaração de variáveis que serão utilizadas no código
        Dim sql As String
        Dim sqlcheck As SqlServerCe.SqlCeDataReader

        sql = "SELECT * FROM Cadastro"

        Dim chkcmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)

        Try
            sqlcheck = chkcmd.ExecuteReader()
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

```

```

While sqlcheck.Read()
    If (NomeTextBox1.Text = sqlcheck("Nome").ToString) Then
        FlagBlockSaving = True
    End If
End While

If FlagBlockSaving = False Then

    'Conexão e envio de mensagem para o SQL

    sql = "INSERT INTO Cadastro" _
        + "(Nome, NomeMae, NomePai, CausaNecEsp)" _
        + "VALUES ('" _
        + NomeTextBox1.Text + "','" _
        + NomeMaeTextBox.Text + "','" _
        + NomePaiTextBox.Text + "','" _
        + CausaNecEspTextBox.Text + "'"")"

    Dim cmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)

    Try
        cmd.ExecuteNonQuery()
        TotalRegistros()
        NumIDTextBox.Enabled = False
        NumIDTextBox.Text = NumRegistros + 1
        FlagSalvar = True

        MsgBox("Cadastro feito com sucesso!")

    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
Else
    MsgBox("Nome já cadastrado!") 'Caso nome já exista

End If
RetornaForm()
End Sub

'Conta o total de registros
Public Sub TotalRegistros()

    Dim strSQL As String = "Select Count(*) From Cadastro"

    Dim cmd As New SqlServerCe.SqlCeCommand(strSQL, ConexaoBD)

```

```

    Try
        NumRegistros = cmd.ExecuteScalar
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

End Sub

'Rotinas para seleção e modificação de cadastros
Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged

    Dim strSql As String = "SELECT * FROM Cadastro WHERE Nome = '" &
ComboBox1.SelectedItem.ToString & "'"
    Dim cmd As New SqlServerCe.SqlCeCommand(strSql, conexaoBD)

    Dim sqlReader As SqlServerCe.SqlCeDataReader

    Try
        sqlReader = cmd.ExecuteReader()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

    NomeTextBox1.Text = sqlReader.GetString(1)
    NomeMaeTextBox.Text = sqlReader.Item("NomeMae").ToString
    NomePaiTextBox.Text = sqlReader.Item("NomePai").ToString
    CausaNecEspTextBox.Text = sqlReader.Item("CausaNecEsp").ToString
    NumIDTextBox.Text = sqlReader.Item("Numero").ToString

End Sub

Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button9.Click

    If ComboBox1.SelectedItem = Nothing Then
        MsgBox("Selecione algum nome para fazer alteração")
    Else
        Dim strSql As String = "UPDATE Cadastro SET (Nome = '" &
NomeTextBox1.ToString & "'" & _
        "UPDATE Cadastro SET (NomeMae = '" &
NomeMaeTextBox.ToString & "'" & _
        "UPDATE Cadastro SET (NomePai = '" &
NomePaiTextBox.ToString & "'" & _
        "UPDATE Cadastro SET (CausaNecEsp = '" &
CausaNecEspTextBox.ToString & "'" &

```

```

        Dim cmd As New SqlServerCe.SqlCeCommand(strSql, ConexaoBD)

        Try
            cmd.ExecuteNonQuery()
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End If
End Sub
End Class

```

- **Cadastro de tarefas**

Na tela de “Cadastro de tarefas” é possível adicionar novas tarefas e suas descrições. Os nomes das tarefas irão aparecer na *groupbox* tarefas presente na tela inicial.

```

Public Class AddTarefa

    Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'Inicia banco de dados

'Me.TarefasDatabaseTableAdapter.Fill(Me.DatabaseTarefasDataSet.Tarefas)

        'Conta número de tarefas e preenche o campo "número". Além disso,
limpa o CheckListBox onde tem o nome
        'dos robôs capazes de executar a tarefa.
        RetornaForm()

        'Define a string de comando e as variáveis de leitura, conexão e
comando do Sql Database.
        'Neste caso a string se refere à seleção de todos os itens do Robos
para contar o número de elementos.
        Dim sql_string As String = "SELECT * FROM Robos"
        Dim strldr As SqlServerCe.SqlCeDataReader
        Dim cmd = New SqlServerCe.SqlCeCommand(sql_string, ConexaoBD)

        'Roda o comando para o cálculo do número de elementos e preenchimento
do CheckListBox. Além disso,

```

```

        'caso haja erro exibe a mensagem correspondente ao erro (em inglês,
pois vem direto do compilador VB).
    Try
        strldr = cmd.ExecuteReader
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

    'Preenche a lista CheckListBox. Só consegue ser executado caso a parte
anterior tenha sido bem sucedida.
    While strldr.Read()
        'CheckedListBox2.Items.Add(strldr("NomeRobo"))
    End While

    TotalRegistros()
    NumTarefaTextBox.Enabled = False
    NumTarefaTextBox.Text = NumRegistros + 1

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Me.Close()
End Sub

Private Sub RetornaForm()
    'Esta rotina serve para carregar o número de tarefas já existentes e
limpar os campos de nome e descrição.
    'Todo o procedimento de conexão com banco de dados e envio de comando
e string são feitos.
    Dim sql As String = "SELECT * FROM Tarefas"
    Dim comm As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)
    Dim streader As SqlServerCe.SqlCeDataReader

    Try
        streader = comm.ExecuteReader
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

    'Limpa alguns campos
    NomeTarefaTextBox.Clear()
    TDescricaoTextBox.Clear()
    'CheckedListBox2.Items.Clear()
    ComboBox1.Items.Clear()

```

```

'Atualização das tarefas para pesquisa
While streader.Read()
    ComboBox1.Items.Add(streader("Nome"))
End While

End Sub

'Rotina para salvar uma nova tarefa
Private Sub Button5_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    'Define flag que impede que duas tarefas sejam salvas com o mesmo nome
    Dim FlagBlockSaving As Boolean = False

    'Define a string para selecionar todos os dados da tabela "Tarefas".
Além disso,
    'também são definidas as variáveis de leitura, conexão e comando.
    Dim sql As String
    Dim sqlcheck As SqlServerCe.SqlCeDataReader

    sql = "SELECT * FROM Tarefas"

    Dim chkcmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)

    Try
        sqlcheck = chkcmd.ExecuteReader()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

    While sqlcheck.Read()
        If (NomeTarefaTextBox.Text = sqlcheck("Nome").ToString) Then
            FlagBlockSaving = True
        End If
    End While

    If FlagBlockSaving = False Then

        sql = "INSERT INTO Tarefas(Numero, Nome, Descricao) VALUES ('" _
            + NumTarefaTextBox.Text + "','" _
            + NomeTarefaTextBox.Text + "','" _
            + TDescricaoTextBox.Text + "')"

        Dim cmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)

        'Caso o bloco anterior seja executado, o comando é realizado e a
nova tarefa é inserida dentro

```

'do banco de dados. Caso algum erro aconteça o sistema exibe uma mensagem de erro (em inglês) e uma outra em português pedindo para inserir outro nome.

```

    Try
        cmd.ExecuteNonQuery()
        MsgBox("Tarefa adicionada com sucesso!")
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
Else
    MsgBox("Já existe uma tarefa cadastrada com este nome. Informe
outro nome.")
End If

RetornaForm()
End Sub

```

'Rotina para ler total de registros

```

Public Sub TotalRegistros()

    Dim strSQL As String = "Select Count(*) From Tarefas"

    Dim cmd As New SqlServerCe.SqlCeCommand(strSQL, ConexaoBD)

    Try
        NumRegistros = cmd.ExecuteScalar
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

End Sub

```

'Rotinas para selecionar e modificar tarefas

```

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged

```

'Define a string para selecionar todos os dados da tabela "Tarefas".
Além disso,

'também são definidas as variáveis de leitura, conexão e comando.

```

Dim sql As String = Nothing
Dim sqlcheck As SqlServerCe.SqlCeDataReader = Nothing

```

```

sql = "SELECT * FROM Tarefas WHERE (Nome = '" &
ComboBox1.SelectedItem.ToString & "')"

```

```

Dim chkcmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)

```

```

Try
    sqlcheck = chkcmd.ExecuteReader()
Catch ex As Exception
    MsgBox(ex.Message)
End Try

'Preencher campos
NumTarefaTextBox.Text = sqlcheck("NumTarefa").Value.ToString
NomeTarefaTextBox.Text = sqlcheck("Nome").Value.ToString
TDescricaoTextBox.Text = sqlcheck("Descricao").Value.ToString

End Sub

'Preencher campos
Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button9.Click

    'Define a string para selecionar todos os dados da tabela "Tarefas".
Além disso, também são definidas as variáveis de leitura, conexão e comando.
    Dim sql As String = Nothing
    Dim sqlcheck As SqlServerCe.SqlCeDataReader

    sql = "SELECT * FROM Tarefas"

    Dim chkcmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)
    Try
        sqlcheck = chkcmd.ExecuteReader()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

    sql = "UPDATE Tarefas(NumTarefa, Nome, Descricao) VALUES ('" _
        + NumTarefaTextBox.Text + "','" _
        + NomeTarefaTextBox.Text + "','" _
        + TDescricaoTextBox.Text + "')"

    Dim cmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)

    'Caso o bloco anterior seja executado, o comando é realizado e a nova
tarefa é inserida dentro
    'do banco de dados. Caso algum erro aconteça o sistema exibe uma
mensagem de erro (em inglês).
    Try
        cmd.ExecuteNonQuery()
        MsgBox("Tarefa modificada com sucesso!")
    Catch ex As Exception

```

```

        MsgBox(ex.Message)
    End Try

    'Rotina que atualiza as tarefas dentro da janela principal de tarefas
    (caso a mesma esteja aberta).
    RetornaForm()
End Sub

End Class

```

- **Cadastro de robôs**

É possível nesta tela realizar o cadastro de robôs e colocar quais comandos são interpretados pelo mesmo para ir para as direções e executar certos movimentos.

```

Public Class Robos

    'Fecha a janela
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        Me.Close()
    End Sub

    'Carrega a tela do Cadastro de Robos
    Private Sub Form7_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the
'DatabaseRobosDataSet.Robos' table. You can move, or remove it, as needed.
        'Me.RoboDatabaseTableAdapter.Fill(Me.DatabaseRobosDataSet.Robos)

        'Define a string de comando e as variáveis de leitura, conexão e
comando do Sql Database.
        'Neste caso a string se refere à seleção de todos os itens do Robos
para contar o número de elementos.
        Dim sql_string As String = "SELECT * FROM Robos"
        Dim strldr As SqlServerCe.SqlCeDataReader
        Dim cmd = New SqlServerCe.SqlCeCommand(sql_string, ConexaoBD)

        'Roda o comando para o cálculo do número de elementos e preenchimento
do CheckListBox. Além disso,
        'caso haja erro exibe a mensagem correspondente ao erro (em inglês,
pois vem direto do compilador VB).

```

```

Try
    strldr = cmd.ExecuteReader
Catch ex As Exception
    MsgBox(ex.Message)
End Try

'Preenche a lista CheckListBox. Só consegue ser executado caso a parte
anterior tenha sido bem sucedida.
While strldr.Read()
    ComboBox3.Items.Add(strldr("NomeRobo").ToString)
End While

End Sub

'Guarda tipo de robô
Private Sub TipoRobot()

    If RadioButton1.Checked = True Then
        Tipo_Robo = "Manipulador"
    End If

    If RadioButton2.Checked = True Then
        Tipo_Robo = "Móvel"
    End If

    If RadioButton3.Checked = True Then
        Tipo_Robo = "Manipulador e móvel"
    End If

    If RadioButton4.Checked = True Then
        Tipo_Robo = "Virtual (Software)"
    End If

End Sub

'Salva no banco de dados
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click

    Dim FlagBlockSaving As Boolean = False

    Dim sql As String = "SELECT * FROM Robos"
    Dim sqlcheck As SqlServerCe.SqlCeDataReader

    Dim chkcmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)

```

```

Try
    sqlcheck = chkcmd.ExecuteReader()
Catch ex As Exception
    MsgBox(ex.Message)
End Try

While sqlcheck.Read()
    If (NomeRoboTextBox.Text = sqlcheck("NomeRobo").ToString) Then
        FlagBlockSaving = True
    End If
End While

If FlagBlockSaving = False Then
    'Salva no banco de dados

    'Este código irá salvar no banco de dados novas inclusões

    'Inserção das portas disponíveis

    TipoRobot() 'Verifica qual o tipo do robô a ser utilizado
    'Declaração de variáveis que serão utilizadas no código

    sql = "INSERT INTO Robos(Nome, DirRight, DirLeft, DirUp, DirDown,
BotaoX, BotaoY, BotaoA, BotaoB, BotaoC, Fabricante) VALUES('" _
        + NomeRoboTextBox.Text + "','" _
        + DirRightTextBox.Text + "','" _
        + DirLeftTextBox.Text + "','" _
        + DirUpTextBox.Text + "','" _
        + DirDownTextBox.Text + "','" _
        + BotaoXTextBox.Text + "','" _
        + BotaoYTextBox.Text + "','" _
        + BotaoATextBox.Text + "','" _
        + BotaoBTextBox.Text + "','" _
        + BotaoCTextBox.Text + "','" _
        + FabricanteTextBox.Text + "'"")

    Dim cmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)

Try
    cmd.ExecuteNonQuery()
    FlagSalvar = True
    MsgBox("Cadastro de robô feito com sucesso!")
Catch ex As Exception
    MsgBox(ex.Message)
End Try

```

```

Else
    MsgBox("Nome de robô já cadastrado.")
End If

End Sub

End Class

```

- **Cadastro de dispositivos auxiliares**

Funciona da mesma forma que o cadastro de robôs. Porém é voltado para os dispositivos auxiliares (sensores). Com esse parte do código junto com o cadastro de robôs é possível fazer a tradução do comando do sensor para o comando compreensível ao robô.

```

Public Class Auxiliares
    'Declaração de Variáveis Globais (dentro desta classe)
    Public FlagDirecao As String
    Public FlagFormat As String
    Public RecordSTR As String

    'Fecha a janela
    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
        Close()
    End Sub

    'Carrega a janela
    Private Sub Auxiliares_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim strSql As String
        Dim strdr As SqlServerCe.SqlCeDataReader

        strSql = "SELECT * FROM Dipositivos_Auxiliares"
        Dim cmd As New SqlServerCe.SqlCeCommand(strSql, ConexaoBD)
        strdr = cmd.ExecuteReader()

        ComboBox3.Items.Clear()
        While (strdr.Read())
            ComboBox3.Items.Add(strdr("Equipamento").ToString)
        End While
    End Sub
End Class

```

```

'Salva no banco de dados
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click

    'Esta parte do código impede que haja dois cadastros com o mesmo nome
    ou número
    Dim FlagBlockSaving As Boolean = False

    'Declaração de variáveis que serão utilizadas no código
    Dim sql As String
    Dim sqlcheck As SqlServerCe.SqlCeDataReader

    sql = "SELECT * FROM Dpositivos_Auxiliares"

    Dim chkcmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)

    Try
        sqlcheck = chkcmd.ExecuteReader()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

    While sqlcheck.Read()
        If (EquipamentoTextBox.Text = sqlcheck("Equipamento").ToString)
Then
            FlagBlockSaving = True
        End If
    End While

    If FlagBlockSaving = False Then

        'Variáveis para conexão e envio de mensagem para o SQL

        sql = "INSERT INTO Dpositivos_Auxiliares" _
            + "(Equipamento, Fabricante, StartByte, Tamanho, StopByte,
DirRight, DirLeft, DirUp, DirDown, BotaoX, BotaoY, BotaoA, BotaoB, BotaoC)" _
            + " VALUES ('" _
            + EquipamentoTextBox.Text + "','" _
            + FabricanteTextBox.Text + "','" _
            + TextBox1.Text + "','" _
            + TextBox2.Text + "','" _
            + TextBox3.Text + "','" _
            + DirRightText + "','" _
            + DirLeftText + "','" _
            + DirUpText + "','" _

```

```

        + DirDownText + "','" _
        + ButtonAText + "','" _
        + ButtonBText + "','" _
        + ButtonCText + "','" _
        + ButtonXText + "','" _
        + ButtonYText + "')"

    Dim cmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)

    Try
        cmd.ExecuteNonQuery()
        FlagSalvar = True

        MsgBox("Cadastro feito com sucesso!")

    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

Else
    MsgBox("Nome já cadastrado!")
End If
End Sub

'Relaciona os movimentos
Private Sub PictureBox5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox5.Click
    FlagDirecao = "Direita"
End Sub

Private Sub PictureBox6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox6.Click
    FlagDirecao = "Trás"
End Sub

Private Sub PictureBox2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox2.Click
    FlagDirecao = "Frente"
End Sub

Private Sub PictureBox4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox4.Click
    FlagDirecao = "Esquerda"
End Sub

```

```

'Guarda a associação dos movimentos e o tipo de variável
(string/integer).

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
    Select Case ComboBox1.SelectedItem.ToString
        Case "Strings"
            FlagFormat = "STR"
            TextBox9.ReadOnly = False
            TextBox7.ReadOnly = True
            TextBox8.ReadOnly = True
            RecordSTR = FlagFormat & "#" & TextBox9.Text.ToString & "-" &
TextBox6.Text.ToString & "-" & TextBox4.Text.ToString

        Case "Maior que"
            FlagFormat = "+MQ"
            TextBox9.ReadOnly = True
            TextBox7.ReadOnly = True
            TextBox8.ReadOnly = False
            RecordSTR = FlagFormat & "#" & TextBox8.Text.ToString & "-" &
TextBox6.Text.ToString & "-" & TextBox4.Text.ToString

        Case "Menor que"
            FlagFormat = "-MQ"
            TextBox9.ReadOnly = True
            TextBox7.ReadOnly = False
            TextBox8.ReadOnly = True
            RecordSTR = FlagFormat & "#" & TextBox7.Text.ToString & "-" &
TextBox6.Text.ToString & "-" & TextBox4.Text.ToString

        Case "Entre valores"
            FlagFormat = "2VL"
            TextBox9.ReadOnly = True
            TextBox7.ReadOnly = False
            TextBox8.ReadOnly = False
            RecordSTR = FlagFormat & "#" & TextBox7.Text.ToString & "-" &
TextBox8.Text.ToString
    End Select

End Sub

'Carrega a tela
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Select Case FlagDirecao
        Case FlagDirecao = "Frente" : auxdev_up = RecordSTR
    End Select
End Sub

```

```

        Case FlagDirecao = "Tras" : auxdev_down = RecordSTR
        Case FlagDirecao = "Direita" : auxdev_right = RecordSTR
        Case FlagDirecao = "Esquerda" : auxdev_left = RecordSTR
    End Select
End Sub
End Class

```

- **Relatórios**

Os relatórios compõe uma parte importante do programa, pois otimizam e organizam todas as informações em uma página HTML.

```

Public Class HTML

    'Carrega a tela
    Private Sub Form5_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'Esta função é responsável por escrever o código HTML que irá compor
        'o relatório. Fica tudo gravado em uma string que depois será
transferida
        'para um arquivo html e pode ser lida em qualquer browser.
    If FlagGeraHTML = True Then
        EstruturaHTML()
        'Variáveis responsáveis por gerar o diretório que será gravado o
relatório
        'o diretório é "<nome da criança> - dia do ano - hora - minuto"
        Dim marcatepo As String = Now.DayOfYear & "-" & Now.Hour & "-" &
Now.Minute
        Dim Url As String = directory & "\" & NomeSelecioneado & " - " &
marcatepo & ".html"

        'Escreve-se o código html no arquivo descrito em cima por Url.
Texto é a variável
        'que guarda o código html gerado pela função EstruturaHTML.
My.Computer.FileSystem.WriteAllText(Url, texto, False)

        'Exibe a mensagem de onde foi gravado o arquivo.
MsgBox(Url)

        'Mostra no Browser do próprio programa.
WebBrowser1.Url = New System.Uri(Url)
    End If

```

```

End Sub

Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
    'Fecha o programa.
    Me.Close()
End Sub

'Monta o código HTML
Private Sub EstruturaHTML()

'Apaga possíveis estruturas HTML
texto = Nothing

'Começa novo relatório
texto = texto & "<html>"
texto = texto & "<img src='" & Application.StartupPath &
"\Imagens\Robotica Educacional.png'">"
texto = texto & "<br>"
texto = texto & "Nome da Criança: " & NomeSelecioneado & "<br>"
texto = texto & "Nome do Teste: " & TesteSelecioneado & "<br>"
texto = texto & "Nome do Robô: " & RoboSelecioneado & "<br>"
texto = texto & "<p>"
texto = texto & "Tempo Gasto: " & TempoCorrido.ToString.Substring(0,
8) & "<br>"
texto = texto & "GAS Score: " & GAS_Score & "<br>"
texto = texto & "<p>"

'Variáveis para contagem de posição e tempo. Além de strings para
'guardar qual diretório será gravado o relatório.
Dim i As Integer = 0
Dim auxi As Integer = 0
Dim posString As Integer = 0
Dim PathStart As String = directory
Dim TimeAtMovimentStr As String = Nothing

'Zera os erros
Erro = 0

'O bloco seguinte do programa faz as seguintes tarefas:
' (1) Verifica se há alguma movimentação. Se sim continua.
' (2) Verifica se a FlagSequenciaChecked está marcada (indicando que
há movimentação correta)

```

```

' (3) Se (2) for verdadeiro analisa se a movimentação está correta e
indica se está certo ou errado
' (4) Se (2) for falso apenas registra os movimentos.

If Movimentacao <> Nothing Then

    If FlagSequenciaChecked Then
        texto = texto & "Sequência Correta:"
        texto = texto & "<p>"

        While auxi < Sequencia.Length

            Select Case Sequencia(auxi)

                Case "w" : texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\OnUpArrow.png', align='middle'">"
                Case "s" : texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\OnDownArrow.png', align='middle'">"
                Case "d" : texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\OnRightArrow.png', align='middle'">"
                Case "a" : texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\OnLeftArrow.png', align='middle'">"
                Case "p" : texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\A.png', align='middle'">"
                Case "g" : texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\B.png', align='middle'">"

            End Select

            auxi = auxi + 1

        End While

        texto = texto & "<p>"
        texto = texto & "Movimentação: "
        texto = texto & "<p>"

    Else

        texto = texto & "Movimentação: "
        texto = texto & "<p>"

    End If

    Dim I_aux As Integer

    While i < Movimentacao.Length

```

```

        I_aux = i
        If Movimentacao(I_aux) = "t" Then
        Else
            Select Case Movimentacao(I_aux)
            Case "w"

                If FlagSequenciaChecked Then
                    If Movimentacao(I_aux) = Sequencia(I_aux -
Erro) Then
                        texto = texto & (posString + 1).ToString
                        texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'">"
                        texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\UpGreen.png', align='middle'">"
                    Else
                        texto = texto & (posString + 1).ToString
                        texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'">"
                        texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\UpRed.png', align='middle'">"
                        Erro = Erro + 1
                    End If
                Else
                    texto = texto & (posString + 1).ToString
                    texto = texto & "<img src='" & directory & "\\"
& (I_aux + 1).ToString & ".bmp', align='middle'">"
                    texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\UpArrow.png', align='middle'">"
                End If
                texto = texto & "Movimentação para frente - "
                texto = texto &
TimeAtMovement(I_aux).ToString.Substring(0, 8) & " :: " & Humor(I_aux) &
"<br>"

                texto = texto & "<p>"
                posString = posString + 1

            Case "s"

                If FlagSequenciaChecked Then
                    If Movimentacao(I_aux) = Sequencia(I_aux -
Erro) Then
                        texto = texto & (posString + 1).ToString
                        texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'">"
                        texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\DownGreen.png', align='middle'">"

```

```

        Else
            texto = texto & (posString + 1).ToString
            texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'>"
            texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\DownRed.png', align='middle'>"
            Erro = Erro + 1
        End If
    Else
        texto = texto & (posString + 1).ToString
        texto = texto & "<img src='" & directory & "\\"
& (I_aux + 1).ToString & ".bmp', align='middle'>"
        texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\DownArrow.png', align='middle'>"
    End If
    texto = texto & "Movimentação para trás - "
    texto = texto &
TimeAtMovement(I_aux).ToString.Substring(0, 8) & " :: " & Humor(I_aux) &
"<br>"

    texto = texto & "<p>"
    posString = posString + 1

Case "d"

    If FlagSequenciaChecked Then
        If Movimentacao(I_aux) = Sequencia(I_aux -
Erro) Then
            texto = texto & (posString + 1).ToString
            texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'>"
            texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\RightGreen.png', align='middle'>"
        Else
            texto = texto & (posString + 1).ToString
            texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'>"
            texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\RightRed.png', align='middle'>"
            Erro = Erro + 1
        End If
    Else
        texto = texto & (posString + 1).ToString
        texto = texto & "<img src='" & directory & "\\"
& (I_aux + 1).ToString & ".bmp', align='middle'>"
        texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\RightArrow.png', align='middle'>"

```

```

        End If
        texto = texto & "Movimentação para direita - "
        texto = texto &
TimeAtMovement(I_aux).ToString.Substring(0, 8) & " :: " & Humor(I_aux) &
"<br>"

        texto = texto & "<p>"
        posString = posString + 1

    Case "a"

        If FlagSequenciaChecked Then
            If Movimentacao(I_aux) = Sequencia(I_aux -
Erro) Then

                texto = texto & (posString + 1).ToString
                texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'">"
                texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\LeftGreen.png', align='middle'">"
            Else
                texto = texto & (posString + 1).ToString
                texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'">"
                texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\LeftRed.png', align='middle'">"
                Erro = Erro + 1
            End If
        Else
            texto = texto & (posString + 1).ToString
            texto = texto & "<img src='" & directory & "\\"
& (I_aux + 1).ToString & ".bmp', align='middle'">"
            texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\LeftArrow.png', align='middle'">"
        End If
        texto = texto & "Movimentação para esquerda - "
        texto = texto &
TimeAtMovement(I_aux).ToString.Substring(0, 8) & " :: " & Humor(I_aux) &
"<br>"

        texto = texto & "<p>"
        posString = posString + 1

    Case "p"

        If FlagSequenciaChecked Then
            If Movimentacao(I_aux) = Sequencia(I_aux -
Erro) Then

                texto = texto & (posString + 1).ToString

```

```

        texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'>"
        texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\A.png', align='middle'>"
    Else
        texto = texto & (posString + 1).ToString
        texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'>"
        texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\A.png', align='middle'>"
        Erro = Erro + 1
    End If
Else
    texto = texto & (posString + 1).ToString
    texto = texto & "<img src='" & directory & "\\"
& (I_aux + 1).ToString & ".bmp', align='middle'>"
    texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\A.png', align='middle'>"
    End If
    texto = texto & "Troca estado da pinça - "
    texto = texto &
TimeAtMovement(I_aux).ToString.Substring(0, 8) & " :: " & Humor(I_aux) &
"<br>"
    texto = texto & "<p>"
    posString = posString + 1

Case "g"

    If FlagSequenciaChecked Then
        If Movimentacao(I_aux) = Sequencia(I_aux -
Erro) Then
            texto = texto & (posString + 1).ToString
            texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'>"
            texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\B.png', align='middle'>"
        Else
            texto = texto & (posString + 1).ToString
            texto = texto & "<img src='" & directory &
"\\" & (I_aux + 1).ToString & ".bmp', align='middle'>"
            texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\B.png', align='middle'>"
            Erro = Erro + 1
        End If
    Else
        texto = texto & (posString + 1).ToString

```

```

        texto = texto & "<img src='" & directory & "\"
& (I_aux + 1).ToString & ".bmp', align='middle'>"
        texto = texto & "<img src='" &
Application.StartupPath & "\\Imagens\\B.png', align='middle'>"
        End If
        texto = texto & "Troca de estado da garra - "
        texto = texto &
TimeAtMovement(I_aux).ToString.Substring(0, 8) & " :: " & Humor(I_aux) &
"<br>"

        texto = texto & "<p>"
        posString = posString + 1

    End Select

End If
i = i + 1

End While
End If

'Por fim grava-se no banco de dados de reat6rio, todas as informa76es
relevantes para a an6lise estatística
Dim sql As String = Nothing
Dim sqlcheck As SqlServerCe.SqlCeDataReader = Nothing

If FlagSequenciaChecked = False Then
    Erro = 0
    Sequencia = "N6o existe"
End If

sql = "INSERT INTO Relatorios(Nome, Tarefas, Robo, Movimentacao,
Acertos, Tempo, Sequencia, Data, Hora, GAS) VALUES ('" _
    & NomeSelecionado & "','" _
    & TesteSelecionado & "','" _
    & RoboSelecionado & "','" _
    & Movimentacao & "','" _
    & Erro & "','" _
    & TempoCorrido.ToString.Substring(0, 8) & "','" _
    & Sequencia & "','" _
    & Now.Date & "','" _
    & Now.Hour & Now.Minute & Now.Second & "','" _
    & GAS_Score & "')"

Dim cmd As New SqlServerCe.SqlCeCommand(sql, ConexaoBD)

Try

```

```

        cmd.ExecuteNonQuery()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

'Janela abrir arquivo HTML
Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel1.LinkClicked

    OpenFileDialog1.AddExtension = True
    OpenFileDialog1.ShowDialog()

    Dim Url As String = OpenFileDialog1.FileName()
    If Url <> "OpenFileDialog1" Then
        WebBrowser1.Url = New System.Uri(Url)
    End If

End Sub
End Class

```

- **Calibração**

Esta tela serve para capturar as informações do sensor e associar os valores enviados pelo mesmo a um movimento. Para isso basta clicar no botão correspondente ao movimento quando a criança já estiver na posição correta com a cabeça (ou o braço) e então esperar a coleta de 10 amostras de sinal. Com isso associa-se o valor médio com o movimento feito.

```

Public Class Calibracao

    'Abre a janela
    Private Sub Calibracao_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        'Criação das nova thread e inicialização
        Dim BTThread As New Threading.Thread(AddressOf
Tarefas.BluetoothThread)
        BTThread.Start()
    End Sub

    'Fecha a janela
    Private Sub Sair_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Sair.Click
        Close()
    End Sub

```

```

'As rotinas seguintes fazem a leitura do sensor e associa os
valores com o movimento correto
Private Sub Frente_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Frente.Click
    Dim media As Integer
    For i As Integer = 1 To 10
        media = media + getAcelerometro0
        System.Threading.Thread.Sleep(500)
    Next
    media = media / 10 'Fazendo a média
    IrFrente = media
    TextBox1.Text = TextBox1.Text & vbCrLf & " >> Frente: " &
media
End Sub

Private Sub Tras_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Tras.Click
    Dim media As Integer
    For i As Integer = 1 To 10
        media = media + getAcelerometro0
        System.Threading.Thread.Sleep(500)
    Next
    media = media / 10 'Fazendo a média
    IrTras = media
    TextBox1.Text = TextBox1.Text & vbCrLf & " >> Trás: " &
media
End Sub

Private Sub Direita_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Direita.Click
    Dim media As Integer
    For i As Integer = 1 To 10
        media = media + getAcelerometro1
        System.Threading.Thread.Sleep(500)
    Next
    media = media / 10 'Fazendo a média
    IrDir = media
    TextBox1.Text = TextBox1.Text & vbCrLf & " >> Direita: " &
media
End Sub

Private Sub Esquerda_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Esquerda.Click
    Dim media As Integer
    For i As Integer = 1 To 10
        media = media + getAcelerometro1
        System.Threading.Thread.Sleep(500)
    Next
    media = media / 10 'Fazendo a média
    IrEsq = media
    TextBox1.Text = TextBox1.Text & vbCrLf & " >> Esquerda: " &
media
End Sub

Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles PictureBox1.Click
    Dim media As Integer
    For i As Integer = 1 To 10
        media = media + EMG_Medio
        System.Threading.Thread.Sleep(500)
    Next

```

```

        Next
        media = media / 10 'Fazendo a média
        IrEsq = media
        TextBox1.Text = TextBox1.Text & vbNewLine & " >> EMG (Limiar):
" & media
    End Sub
End Class

```

- **FastCalibration**

Janela criada para rapidamente mudar os valores de *threshold* do sensor inercial. Assim pode-se aumentar ou diminuir a sensibilidade da placa para direita, esquerda, frente e trás, além do *threshold* do sinal EMG.

```

Public Class Calibracao

    'Fecha janela
    Private Sub Sair_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Sair.Click
        Close()
    End Sub

    'Carrega tela
    Private Sub Calibracao_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Criação das nova thread e inicialização
        Dim BTThread As New Threading.Thread(AddressOf
Tarefas.BluetoothThread)

        BTThread.Start()
    End Sub

    'Rotinas a seguir recalibram o sensor inercial de forma rápida
(inserindo valores)
    Private Sub Frente_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Frente.Click
        Dim media As Integer
        For i As Integer = 1 To 10
            media = media + getAcelerometro0
            System.Threading.Thread.Sleep(500)
        Next
        media = media / 10 'Fazendo a média
        IrFrente = media
        TextBox1.Text = TextBox1.Text & vbNewLine & " >> Frente: " & media
    End Sub
End Class

```

```

End Sub

Private Sub Tras_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Tras.Click
    Dim media As Integer
    For i As Integer = 1 To 10
        media = media + getAcelerometro0
        System.Threading.Thread.Sleep(500)
    Next
    media = media / 10 'Fazendo a média
    IrTras = media
    TextBox1.Text = TextBox1.Text & vbCrLf & " >> Trás: " & media
End Sub

Private Sub Direita_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Direita.Click
    Dim media As Integer
    For i As Integer = 1 To 10
        media = media + getAcelerometro1
        System.Threading.Thread.Sleep(500)
    Next
    media = media / 10 'Fazendo a média
    IrDir = media
    TextBox1.Text = TextBox1.Text & vbCrLf & " >> Direita: " & media
End Sub

Private Sub Esquerda_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Esquerda.Click
    Dim media As Integer
    For i As Integer = 1 To 10
        media = media + getAcelerometro1
        System.Threading.Thread.Sleep(500)
    Next
    media = media / 10 'Fazendo a média
    IrEsq = media
    TextBox1.Text = TextBox1.Text & vbCrLf & " >> Esquerda: " & media
End Sub

Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox1.Click
    Dim media As Integer
    For i As Integer = 1 To 10
        media = media + SEMG_Medio
        System.Threading.Thread.Sleep(500)
    Next

```

```

        media = media / 10 'Fazendo a média
        IrEsq = media
        TextBox1.Text = TextBox1.Text & vbCrLf & " >> SEMG (Limiar): " &
media
    End Sub
End Class

```

- **Seqüência**

A seqüência correta pode ser utilizada para forçar a criança a fazer os movimentos corretos, já que o robô só se movimenta com a criança fazendo o movimento correto. Caso contrário o computador registra o movimento (marca como errado) e não envia sinal para o robô.

```

Public Class Sequence

    'Carrega tela
    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Sequencia = ""
        Label5.Hide()
        LimpaFiguras()
        PictureNumber = 0
    End Sub

    'Exibe mensagem de seqüência gravada.
    Private Sub PictureBox5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox5.Click
        MsgBox("A seqüência foi gravada.")
        Tarefas.LinkLabel3.Hide()
        FlagSequenciaChecked = True
        Label5.Show()
    End Sub

    'Rotinas seguintes tratam as figuras e montam a sequencia
    Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox1.Click
        AtualizaFigura(UpArrow)
        Sequencia = Sequencia & "w"
    End Sub

```

```

Private Sub PictureBox3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox3.Click
    AtualizaFigura(DownArrow)
    Sequencia = Sequencia & "s"
End Sub

Private Sub PictureBox2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox2.Click
    AtualizaFigura(RightArrow)
    Sequencia = Sequencia & "d"
End Sub

Private Sub PictureBox4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox4.Click
    AtualizaFigura(LeftArrow)
    Sequencia = Sequencia & "a"
End Sub

'Trata as figuras (atualiza)
Private Sub AtualizaFigura(ByVal PictureBox As String)

    If PictureBoxNumber > 54 Then
        LimpaFiguras()
    End If

    Select Case PictureBoxNumber

        'Primeira Linha de 1 até 18
        Case 0 : P1.Image = Image.FromFile(PictureBox)
        Case 1 : P2.Image = Image.FromFile(PictureBox)
        Case 2 : P3.Image = Image.FromFile(PictureBox)
        Case 3 : P4.Image = Image.FromFile(PictureBox)
        Case 4 : P5.Image = Image.FromFile(PictureBox)
        Case 5 : P6.Image = Image.FromFile(PictureBox)
        Case 6 : P7.Image = Image.FromFile(PictureBox)
        Case 7 : P8.Image = Image.FromFile(PictureBox)
        Case 8 : P9.Image = Image.FromFile(PictureBox)
        Case 9 : P10.Image = Image.FromFile(PictureBox)
        Case 10 : P11.Image = Image.FromFile(PictureBox)
        Case 11 : P12.Image = Image.FromFile(PictureBox)
        Case 12 : P13.Image = Image.FromFile(PictureBox)
        Case 13 : P14.Image = Image.FromFile(PictureBox)
        Case 14 : P15.Image = Image.FromFile(PictureBox)
        Case 15 : P16.Image = Image.FromFile(PictureBox)
        Case 16 : P17.Image = Image.FromFile(PictureBox)
    End Select
End Sub

```

```
Case 17 : P18.Image = Image.FromFile(PictureString)

'Segunda Linha de 19 até 36
Case 18 : P19.Image = Image.FromFile(PictureString)
Case 19 : P20.Image = Image.FromFile(PictureString)
Case 20 : P21.Image = Image.FromFile(PictureString)
Case 21 : P22.Image = Image.FromFile(PictureString)
Case 22 : P23.Image = Image.FromFile(PictureString)
Case 23 : P24.Image = Image.FromFile(PictureString)
Case 24 : P25.Image = Image.FromFile(PictureString)
Case 25 : P26.Image = Image.FromFile(PictureString)
Case 26 : P27.Image = Image.FromFile(PictureString)
Case 27 : P28.Image = Image.FromFile(PictureString)
Case 28 : P29.Image = Image.FromFile(PictureString)
Case 29 : P30.Image = Image.FromFile(PictureString)
Case 30 : P31.Image = Image.FromFile(PictureString)
Case 31 : P32.Image = Image.FromFile(PictureString)
Case 32 : P33.Image = Image.FromFile(PictureString)
Case 33 : P34.Image = Image.FromFile(PictureString)
Case 34 : P35.Image = Image.FromFile(PictureString)
Case 35 : P36.Image = Image.FromFile(PictureString)

'Segunda Linha de 37 até 54
Case 36 : P37.Image = Image.FromFile(PictureString)
Case 37 : P38.Image = Image.FromFile(PictureString)
Case 38 : P39.Image = Image.FromFile(PictureString)
Case 39 : P40.Image = Image.FromFile(PictureString)
Case 40 : P41.Image = Image.FromFile(PictureString)
Case 41 : P42.Image = Image.FromFile(PictureString)
Case 42 : P43.Image = Image.FromFile(PictureString)
Case 43 : P44.Image = Image.FromFile(PictureString)
Case 44 : P45.Image = Image.FromFile(PictureString)
Case 45 : P46.Image = Image.FromFile(PictureString)
Case 46 : P47.Image = Image.FromFile(PictureString)
Case 47 : P48.Image = Image.FromFile(PictureString)
Case 48 : P49.Image = Image.FromFile(PictureString)
Case 49 : P50.Image = Image.FromFile(PictureString)
Case 50 : P51.Image = Image.FromFile(PictureString)
Case 51 : P52.Image = Image.FromFile(PictureString)
Case 52 : P53.Image = Image.FromFile(PictureString)
Case 53 : P54.Image = Image.FromFile(PictureString)

End Select
PictureNumber = PictureNumber + 1
Label1.Text = PictureNumber & " movimentos registrados"

End Sub
```

```
Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
    Me.Close()
End Sub

Private Sub LimpaFiguras()

    'Imagem de 1 até 18
    P1.Image = Nothing
    P2.Image = Nothing
    P3.Image = Nothing
    P4.Image = Nothing
    P5.Image = Nothing
    P6.Image = Nothing
    P7.Image = Nothing
    P8.Image = Nothing
    P9.Image = Nothing
    P10.Image = Nothing
    P11.Image = Nothing
    P12.Image = Nothing
    P13.Image = Nothing
    P14.Image = Nothing
    P15.Image = Nothing
    P16.Image = Nothing
    P17.Image = Nothing
    P18.Image = Nothing

    'Imagem de 19 até 36
    P19.Image = Nothing
    P20.Image = Nothing
    P21.Image = Nothing
    P22.Image = Nothing
    P23.Image = Nothing
    P24.Image = Nothing
    P25.Image = Nothing
    P26.Image = Nothing
    P27.Image = Nothing
    P28.Image = Nothing
    P29.Image = Nothing
    P30.Image = Nothing
    P31.Image = Nothing
    P32.Image = Nothing
    P33.Image = Nothing
    P34.Image = Nothing
    P35.Image = Nothing
    P36.Image = Nothing
```

```

    'Imagem de 19 até 36
    P37.Image = Nothing
    P38.Image = Nothing
    P39.Image = Nothing
    P40.Image = Nothing
    P41.Image = Nothing
    P42.Image = Nothing
    P43.Image = Nothing
    P44.Image = Nothing
    P45.Image = Nothing
    P46.Image = Nothing
    P47.Image = Nothing
    P48.Image = Nothing
    P49.Image = Nothing
    P50.Image = Nothing
    P51.Image = Nothing
    P52.Image = Nothing
    P53.Image = Nothing
    P54.Image = Nothing

    End Sub
End Class

```

- **SplashScreen**

A tela de SplashScreen serve para exibir o logotipo do programa de robótica educacional antes de abrir o menu principal.

```

Public NotInheritable Class SplashScreen

    'TODO: This form can easily be set as the splash screen for the
    application by going to the "Application" tab
    ' of the Project Designer ("Properties" under the "Project" menu).

    Private Sub SplashScreen1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        'Set up the dialog text at runtime according to the application's
assembly information.

```

```

        'TODO: Customize the application's assembly information in the
"Application" pane of the project
        ' properties dialog (under the "Project" menu).

        'Application title

        'Format the version information using the text set into the Version
control at design time as the
        ' formatting string. This allows for effective localization if
desired.
        ' Build and revision information could be included by using the
following code and changing the
        ' Version control's designtime text to "Version {0}.{1:00}.{2}.{3}"
or something similar. See
        ' String.Format() in Help for more information.
        '
        ' Version.Text = System.String.Format(Version.Text,
My.Application.Info.Version.Major, My.Application.Info.Version.Minor,
My.Application.Info.Version.Build, My.Application.Info.Version.Revision)

        'Copyright info
    End Sub

End Class

```

- **AboutBox**

Exibe a tela “sobre... ou *about...*” do programa.

```

Public NotInheritable Class AboutBox2

    Private Sub AboutBox2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        ' Set the title of the form.
        Dim ApplicationTitle As String
        If My.Application.Info.Title <> "" Then
            ApplicationTitle = My.Application.Info.Title
        Else
            ApplicationTitle =
System.IO.Path.GetFileNameWithoutExtension(My.Application.Info.AssemblyName)
        End If
    End Sub
End Class

```

```
        Me.Text = String.Format("About {0}", ApplicationTitle)
        ' Initialize all of the text displayed on the About Box.
        ' TODO: Customize the application's assembly information in the
"Application" pane of the project
        '   properties dialog (under the "Project" menu).
        Me.LabelProductName.Text = My.Application.Info.ProductName
        Me.LabelVersion.Text = String.Format("Version {0}",
My.Application.Info.Version.ToString)
        Me.LabelCopyright.Text = My.Application.Info.Copyright
        Me.LabelCompanyName.Text = My.Application.Info.CompanyName
        Me.TextBoxDescription.Text = My.Application.Info.Description
    End Sub

    Private Sub OKButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OKButton.Click
        Me.Close()
    End Sub
End Class
```

Apêndice II: Programação feita no PIC (POB-BOT)

- **Versão simplificada (sem sEMG)** – utilizada nos testes

```

#include <16f877a.h>
#fuses HS, NOWDT, NOPROTECT, PUT
#use delay(clock = 20000000)

#use fast_io(a)
#use fast_io(b)
#use fast_io(c)
#use fast_io(d)
#use fast_io(e)

#use rs232 (baud=115200, xmit=PIN_C6, parity=N, rcv=PIN_C7)

//Inserere defines sobre liga e desliga
#define lig output_high
#define des output_low

//Inserere defines sobre movimentação
#define parado {des(PIN_D1); des(PIN_D0); des(PIN_D3); des(PIN_D2);}
#define frente {des(PIN_D1); lig(PIN_D0); des(PIN_D3); lig(PIN_D2);}
#define tras {lig(PIN_D1); des(PIN_D0); lig(PIN_D3); des(PIN_D2);}
#define esquerda {des(PIN_D1); lig(PIN_D0); des(PIN_D3); des(PIN_D2);}
#define direita {des(PIN_D1); des(PIN_D0); des(PIN_D3); lig(PIN_D2);}

//Inserindo variáveis
char dado = '#';
short flag = 0;
int i;

#int_rda
void int_receber()
{
    dado = getc();
}

void main()
{
    //Configurações iniciais do PIC
    disable_interrupts (GLOBAL);

    //Manobra anti-bug

```

```
output_low(PIN_D3);

//Mapeando portas como entrada e saída
set_tris_a (0b11111111);
set_tris_b (0b11111111);
set_tris_c (0b10111001);
set_tris_d (0b11110000);
set_tris_e (0b11111111);

//Colocando o robô parado
parado

//Reativando as interrupções
enable_interrupts (GLOBAL);
enable_interrupts (INT_RDA);

while (true)
{
    switch (dado)
    {
        //Associando caracteres às direções
        case 'w': frente; dado = '#'; break;
        case 's': tras; dado = '#'; break;
        case 'a': esquerda; dado = '#'; break;
        case 'd': direita; dado = '#'; break;

        //Associando ao movimento de parar
        case 't': parado; dado = '#'; break;
    }

    //Aguarda um tempo para o processamento do próximo dado recebido
    delay_ms(50); //Aguarda 50ms
}
}
```

Apêndice III: Recursos extras do robô

Alguns recursos do robô ou foram substituídos para uma maior facilidade de uso ou não foram usados. O POB-BOT possui mais duas placas, além das que foram utilizadas.

- **POB-LCD – Display LCD**

O POB-BOT possui um display LCD que se comunica com a placa principal através do conector HE10, localizado na parte esquerda da placa POB-PROTO e na parte inferior do POB-LCD [17]. Utilizando um software adequado para a programação pode-se exibir texto e figuras monocromáticas (inclusive imagens capturadas pela câmera). A figura 5 mostra o POB-LCD e seus componentes.

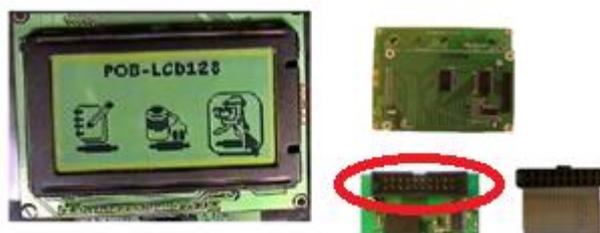


Figura 38 - Placa POB-LCD [17]. Observe o detalhe do cabo HE10 para conexão com a placa principal POB-PROTO (destacado pela elipse vermelha).

- **POB-EYE – Placa da câmera do robô**

Para poder ter capacidade de capturar imagens, o POB-BOT conta com uma câmera em sua parte frontal capaz de obter imagens coloridas com tamanho de 88 x 120 pixels, sendo que cada pixel tem três bytes para cores – 1 para cada componente [17].

Assim como a placa anterior, esta também faz conexão através do cabo HE10 e da conexão I2C bus e então pode ser totalmente integrada às outras placas do robô [17]. Na figura 38 pode-se ver com mais detalhes a câmera.

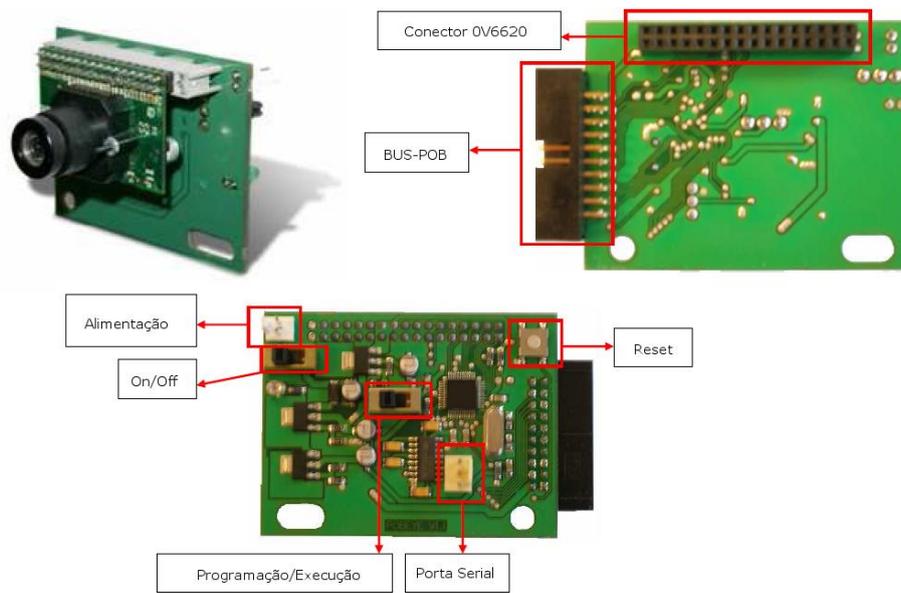


Figura 39 - Placa POB-EYE [17].

Apêndice IV: Carta de aprovação do Comitê de Ética



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
COMITÊ DE ÉTICA EM PESQUISA DO
CENTRO DE CIÊNCIAS DA SAÚDE

Vitória-ES, 09 de dezembro de 2010.

Da: Profa. Dr^a. Ethel Leonor Noia Maciel
 Coordenadora do Comitê de Ética em Pesquisa do Centro de Ciências da Saúde

Para: Prof. (a) Teodiano Freire Bastos Filho
 Pesquisador (a) Responsável pelo Projeto de Pesquisa intitulado: **"Uso da robótica para exploração e descoberta do ambiente para crianças com deficiências físicas severas"**.

Senhor (a) Pesquisador (a),

Informamos a Vossa Senhoria, que o Comitê de Ética em Pesquisa do Centro de Ciências da Saúde da Universidade Federal do Espírito Santo, após analisar o Projeto de Pesquisa nº. **213/10** intitulado: **"Uso da robótica para exploração e descoberta do ambiente para crianças com deficiências físicas severas"** e o **Termo de Consentimento Livre e Esclarecido**, cumprindo os procedimentos internos desta Instituição, bem como as exigências das Resoluções 196 de 10.10.96, 251 de 07.08.97 e 292 de 08.07.99, **APROVOU** o referido projeto, em Reunião Ordinária realizada em 27 de outubro de 2010.

Gostaríamos de lembrar que cabe ao pesquisador responsável elaborar e apresentar os relatórios parciais e finais de acordo com a resolução do Conselho Nacional de Saúde nº 196 de 10/10/96, inciso IX.2. letra "c".

Atenciosamente,



Prof.^a Dr.^a Ethel Leonor Noia Maciel
 COORDENADORA
 Comitê de Ética em Pesquisa
 Centro de Ciências da Saúde/UFES

Comitê de Ética em Pesquisa do Centro de Ciências da Saúde
 Av. Marechal Campos, 1468 – Maruípe – Vitória – ES – CEP 29.040-091.
 Telefax: (27) 3335 7504

Figura 40 - Carta de aprovação do comitê de ética.

Apêndice V: Carta para o Comitê de Ética da Universidade Federal do Espírito Santo

**UNIVERSIDADE FEDERAL DO ESPIRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROF. TEODIANO FREIRE BASTOS FILHO**

Vitória, 10 de Setembro de 2010.

Ao
Comitê de Ética da UFES
Vitória-ES

Prezados Senhores,

O Departamento de Engenharia Elétrica da UFES está desenvolvendo um Projeto de Pesquisa intitulado “Uso da Robótica para Exploração e Descoberta do Ambiente para Crianças com Deficiências Físicas Severas”, financiado pelo FACITEC/PMV, que tem por objetivo investigar o uso da robótica para possibilitar que crianças com deficiências físicas severas sejam capazes de utilizar robôs para exploração e interação com o mundo.

No estágio atual da pesquisa, serão realizados experimentos com crianças com deficiência física, que serão os beneficiários do sistema desenvolvido. Assim, necessitamos da autorização do Comitê de Ética da UFES para a realização dos referidos experimentos com seres humanos.

Ressalto que os voluntários desta pesquisa participarão de sessões de experimentos, em que será pedida a realização de várias tarefas como desenhar e mover objetos utilizando um robô controlado pelos movimentos de cabeça e sinais miográficos captados por eletrodos na face ou no braço. Ressalto também os experimentos não representam qualquer tipo de risco para os voluntários, posto que nos experimentos será utilizado um eletromiógrafo

(modelo EMG800, da empresa EMG Systems do Brasil Ltda.), que atende a todos os requerimentos de segurança, além de dispositivos portáteis de captura de sinais mioelétricos, ambos utilizando eletrodos de superfície. Além disso, é importante citar que em todo o processo não há nenhum procedimento invasivo, visto que os eletrodos são colocados diretamente sobre a pele, sem qualquer perfuração na pele.

Os detalhes do sistema desenvolvido e dos experimentos que serão realizados encontram-se nas próximas páginas deste documento.

Certos de contar com a autorização do Comitê de Ética da UFES para a realização dos referidos experimentos, despedimo-nos mui atenciosamente.

Prof. Dr. Teodiano Freire Bastos Filho
Departamento de Engenharia Elétrica
Universidade Federal do Espírito Santo
Tel.: (27) 4009 20 77
Fax: (27) 4009 27 37
E-mail: teodiano@ele.ufes.br

Apêndice VI: Formulário de Consentimento dos Responsáveis pelos Voluntários

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

INICIAIS DO PACIENTE: _____

DATA DE NASCIMENTO: _____

RESPONSÁVEL: _____

ENDEREÇO: _____

TELEFONE: _____

Eu, _____

portador(a) do registro e identidade número _____, responsável pelo paciente _____, declaro que li e concordo com as afirmações abaixo relacionadas:

1. Título da Pesquisa: Uso da Robótica para Exploração e Descoberta do Ambiente para Crianças com Deficiências Físicas Severas.
2. Pesquisadores Responsáveis: Dr. Teodiano Freire Bastos Filho, Eng. Eletricista Carlos Torturella Valadão (estudante de Mestrado em Engenharia Elétrica) e Eng. Eletricista Magdo Bôrtole (estudante de Mestrado em Engenharia Elétrica).
3. A Pesquisa consiste na utilização de um robô para auxiliar pessoas com deficiência física a terem uma maior interação com o mundo. Será pedido aos voluntários que façam algumas tarefas com o robô, sendo que este será comandado via sensores colocados em um boné que irá vestir, e também via eletrodos que captarão sinais mioelétricos.
4. Os equipamentos utilizados são sensores localizados em um boné que captam a inclinação da cabeça, bem como um sensor mioelétrico capaz de detectar contração muscular, cujos eletrodos estarão localizados no braço ou na região da face próxima aos olhos. Tais dados captados pelos sensores serão utilizados para comandar a movimentação de um robô móvel a rodas, e, conseqüentemente, permite a utilização do mesmo para a realização de tarefas, como desenhar e movimentar objetos no espaço.

5. Não existe qualquer risco no uso destes sensores e eletrodos, bem como a pesquisa não apresenta riscos à saúde dos voluntários. É de grande importância ressaltar que os eletrodos não são invasivos e não dão choques.
6. Terei direito a desistir de participar da pesquisa a qualquer momento sem que isto traga prejuízos a mim ou à pessoa sob a minha responsabilidade.
7. Terei direito a todas as informações pertinentes à pesquisa, mesmo que isto comprometa a minha participação na mesma.
8. Autorizo a divulgação e publicação dos resultados dos exames exclusivamente para fins acadêmicos e científicos.

Confirmo que li e entendi todas as instruções que me foram repassadas pelos coordenadores desta pesquisa e, portanto, dou meu consentimento livre e esclarecido para participar da mesma.

Vitória, ____ de _____ de 2010.

Paciente ou Responsável

Pesquisador Responsável

Apêndice VII: Convite para Participação do Projeto de Pesquisa

CONVITE PARA PARTICIPAR DE UM PROJETO DE PESQUISA

INFORMAÇÃO SOBRE O PROJETO DE PESQUISA

Título do Projeto de Pesquisa:

- Uso da Robótica para Exploração e Descoberta do Ambiente para Crianças com Deficiências Físicas Severas

Pesquisadores Envolvidos:

- Prof. Dr. Teodiano Freire Bastos Filho (Coordenador do Projeto de Pesquisa, Tel.: 4009-2077)
- Carlos Valadão, Engenheiro Eletricista (Estudante de Mestrado em Engenharia Elétrica) Tel.: 4009-2661
- Magdo Bôrtole, Engenheiro Eletricista (Estudante de Mestrado em Engenharia Elétrica), Tel.: 4009-2661

Prezado(a) Senhor(a),

Você está sendo convidado a participar de um Projeto de Pesquisa que será realizada pelo Laboratório de Automação Inteligente (LAI) da UFES.

Este formulário descreve o Projeto de Pesquisa em uma linguagem direta. Por favor, leia este formulário cuidadosamente e certifique que você entendeu o seu conteúdo

antes de decidir participar. Se você tiver qualquer dúvida em relação ao Projeto, por favor, pergunte a um dos pesquisadores do mesmo.

Quem está envolvido neste Projeto de Pesquisa? Por que ele está sendo realizado?

Neste Projeto de Pesquisa estão envolvidos o Prof. Dr. Teodiano Freire Bastos Filho, do Departamento de Engenharia Elétrica da UFES, Coordenador do Projeto, Carlos Torturella Valadão, estudante de Mestrado do Programa de Pós-Graduação em Engenharia Elétrica da UFES e Magdo Bôrtole, estudante de Mestrado do Programa de Pós-Graduação em Engenharia Elétrica da UFES, todos orientados pelo Prof. Dr. Teodiano Freire Bastos Filho.

Este Projeto de Pesquisa está sendo realizado para verificar como robôs podem ser utilizados para auxiliar crianças com deficiência física severa a se comunicar com o meio em que vivem e interagir com as pessoas a seu redor.

Por que você foi escolhido?

Você foi escolhido porque atende a todos os critérios necessários para que possa participar deste Projeto de Pesquisa.

Sobre o quê é o Projeto de Pesquisa? Qual o objetivo da pesquisa?

O Projeto de Pesquisa é um estudo sobre a interação de robôs com humanos, bem como seu uso na área educacional, auxiliando pessoas portadoras de deficiência física severa a interagirem com o meio em que vivem e conseguirem ter mais independência em algumas tarefas.

O objetivo da pesquisa é fazer um sistema robótico capaz de auxiliar pessoas com deficiência física severa a interagir com o meio em que vivem, e, em consequência melhorar a capacidade de socialização, bem como aspectos cognitivos, uma vez que o sistema trará mais independência para as crianças agirem por conta própria.

Se eu concordar em participar, o que eu preciso fazer?

A pesquisa consiste em avaliar o uso de robótica para auxiliar pessoas com deficiência física severa a se comunicar e interagir melhor com o meio em que vive. Para isso, primeiramente será colocado na pessoa um boné que possui uma placa eletrônica com sensores de inclinação e sensores miográficos (estes possuem eletrodos, os quais serão ligados no braço ou, se não for possível movimentação dos membros superiores, na região da face próxima ao olho). É importante ressaltar que este equipamento não dá choques elétricos e não é invasivo.

Em seguida será pedido para a pessoa que está com o boné movimentar o robô, utilizando movimentos de cabeça, e movimentar o braço ou realizar uma piscada de olho, dependendo de onde o eletrodo se localiza. Tais movimentos serão transmitidos para o robô que irá se movimentar de acordo com os movimentos feitos pela pessoa.

Este robô pode realizar várias tarefas. Uma vez que é um robô móvel a rodas dotado de uma pinça, o mesmo pode agarrar objetos e colocá-los em outro lugar, andar sobre uma determinada trajetória e também pode desenhar. Esta última é a principal tarefa, uma vez que permitirá uma melhor comunicação da pessoa com o mundo, comunicação esta feita através do desenho.

Todo o processo será gravado em vídeo e fotografias, porém o mesmo não será utilizado, em momento algum, para identificá-lo.

Quais são os riscos envolvidos na minha participação?

Não existe qualquer risco em utilizar este sistema. Os eletrodos são descartáveis, não machucam, são colocados sobre a pele (não são invasivos) e não dão choques. Os sensores de inclinação também não oferecem qualquer risco, pois apenas captam movimentações da cabeça e também não são invasivos.

Quais são os benefícios associados com a minha participação?

Ao participar deste Projeto de Pesquisa você estará contribuindo para um melhor entendimento de como os robôs podem ser utilizados para auxiliar as pessoas com deficiência física severa a se comunicar com o meio em que vivem, bem como auxiliá-las a terem uma maior independência com o uso destes robôs.

O que acontecerá com a informação que eu fornecerei?

As informações que você fornecer durante a realização das pesquisas não será utilizada para que você seja identificado, a não ser por decisão judicial ou por uma ordem expressa sua. Os vídeos servirão apenas para complementar os dados de pesquisa, e tais imagens não serão divulgadas em momento algum.

Quais são os meus direitos como participante?

Os direitos que você tem como participante desta pesquisa são os seguintes:

- 1) O direito de desistir de participar, em qualquer momento, sem prejuízo para mim.

- 2) O direito de ter qualquer informação sobre a minha participação destruída, se eu considerar que ela pode ser utilizada para me identificar ou se eu considerar que os experimentos representam riscos para mim.

- 3) O direito de ter todas as minhas dúvidas esclarecidas, em qualquer momento.

Quem eu deveria procurar, se eu tiver qualquer dúvida?

- Teodiano Freire Bastos Filho (Tel.:4009-2077, E-mail:teodiano@ele.ufes.br)
- Carlos Torturella Valadão (Tel.:4009-2661, E-mail:carlostvaladao@uol.com.br)
- Magdo Bôrtole (Tel.:4009-2661, E-mail:mgbortole@yahoo.com.br)

Sobre quais outros aspectos eu deveria ser informado se eu decidir participar?

Esta pesquisa não apresenta qualquer risco para os participantes, pois envolve o uso de eletrodos e sensores não invasivos.

Além disso, os experimentos não envolvem qualquer procedimento invasivo, pois o equipamento que será utilizado para adquirir os sinais eletromiográficos é um equipamento comercial que atende a requerimentos de segurança para os usuários.

Atenciosamente,

Prof. Dr. Teodiano Freire Bastos
(Coordenador do Projeto de Pesquisa)

Carlos Torturella Valadão (Estudante de Mestrado)

Magdo Bôrtole (Estudante de Mestrado)

Apêndice VIII: Documentação para o Conselho Nacional de Saúde e CONEP



MINISTÉRIO DA SAÚDE

Conselho Nacional de Saúde

Comissão Nacional de Ética em Pesquisa - CONEP

FOLHA DE ROSTO PARA PESQUISA ENVOLVENDO SERES HUMANOS (versão outubro/99)

1. Projeto de Pesquisa:				
<p>Uso da Robótica para Exploração e Descoberta do Ambiente para Crianças com Deficiências Físicas Severas</p>				
2. Área do Conhecimento (Ver relação no verso)		3. Código: 3.04		4. Nível: (Só áreas do conhecimento 4)
Engenharia Elétrica				
5. Área(s) Temática(s) Especial (s) (Ver fluxograma no verso)		6. Código(s):		7. Fase: (Só área temática 3)
				I () II ()
				III () IV ()
8. Unitermos: (3 opções)				
<p>SUJEITOS DA PESQUISA</p>				
9. Número de sujeitos		10. Grupos Especiais : <18 anos () Portador de Deficiência Mental () Embrião /Feto () Relação de Dependência (Estudantes , Militares, Presidiários, etc) () Outros (X) Não se aplica ()		
No Centro : CREFES		Pessoas de todas as idades, todos cadeirantes.		
Total: 10				
<p>PESQUISADOR RESPONSÁVEL</p>				
11. Nome: Teodiano Freire Bastos Filho				
12. Identidade:		13. CPF.:		19.Endereço (Rua, n.º):
1.351.567		832.812.077-15		Apto. 1303A, Mata da Praia
				Rua Izaltino Marques, 33

14. Nacionalidade: Brasileiro	15. Profissão: Engenheiro Eletricista	20. CEP: 29065-450	21. Cidade: Vitória	22. U.F. ES
16. Maior Titulação: Doutorado	17. Cargo Professor Associado II	23. Fone: 3315-1247 / 4009-2077	24. Fax 4009-2644	
18. Instituição a que pertence: Universidade Federal do Espírito Santo – Departamento de Engenharia Elétrica		25. Email: teodiano@ele.ufes.br		
<p>Termo de Compromisso: Declaro que conheço e cumprirei os requisitos da Res. CNS 196/96 e suas complementares. Comprometo-me a utilizar os materiais e dados coletados exclusivamente para os fins previstos no protocolo e a publicar os resultados sejam eles favoráveis ou não. Aceito as responsabilidades pela condução científica do projeto acima.</p> <p>Data: 10/09/2010 _____</p> <p style="text-align: center;">Assinatura</p>				
INSTITUIÇÃO ONDE SERÁ REALIZADO				
26. Nome: CREFES-Centro de Reabilitação Física do Estado do Espírito Santo		29. Endereço (Rua, nº): Rua Gastão Roubach, s/n Praia da Costa		
27. Unidade/Órgão: Secretaria de Estado da Saúde	30. CEP: 29101-020	31. Cidade: Vila Velha	32. U.F. ES	
28. Participação Estrangeira: Sim () Não (X)		33. Fone:	34. Fax.:	
35. Projeto Multicêntrico: Sim () Não (X) Nacional () Internacional () (Anexar a lista de todos os Centros Participantes no Brasil)				
<p>Termo de Compromisso (do responsável pela instituição): Declaro que conheço e cumprirei os requisitos da Res. CNS 196/96 e suas Complementares e como esta instituição tem condições para o desenvolvimento deste projeto, autorizo sua execução</p> <p>Nome: _____ Cargo: Diretor do CREFES</p> <p>Data: 10/09/2010 _____</p> <p style="text-align: center;">Assinatura</p>				

PATROCINADOR				Não se aplica (X)	
36. Nome:		39. Endereço			
37. Responsável:		40. CEP:	41. Cidade:	42. UF	
38. Cargo/Função:		43. Fone:	44. Fax:		
COMITÊ DE ÉTICA EM PESQUISA - CEP					
45. Data de Entrada: ___/___/___	46. Registro no CEP:	47. Conclusão: Aprovado () Data: ___/___/___		48. Não Aprovado () Data: ___/___/___	
49. Relatório(s) do Pesquisador responsável previsto(s) para: Data: ___/___/___ Data: ___/___/___					
Encaminhado a CONEP: 50. Os dados acima para registro () 51. O projeto para apreciação () 52. Data: ___/___/___		53. Coordenador/Nome _____ Assinatura		Anexar o parecer consubstanciado	
COMISSÃO NACIONAL DE ÉTICA EM PESQUISA - CONEP					
54. Nº Expediente :		56. Data Recebimento :		57. Registro na CONEP:	
55. Processo :					
58. Observações:					