Arcabouço de Software para Auxiliar no Desenvolvimento de Sistemas Interface Cérebro-Computador Pervasivos

Taciana Saad Rached

Dissertação de Mestrado submetida à Coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Angelo Perkusich, D.Sc.
Orientador

Campina Grande, Paraíba, Brasil

Arcabouço de Software para Auxiliar no Desenvolvimento de Sistemas Interface Cérebro-Computador Pervasivos

Taciana Saad Rached

Dissertação de Mestrado apresentada em Junho de 2010

Angelo Perkusich, D.Sc.
Orientador

Hyggo Oliveira de Almeida, D.Sc.
Examinador
Maria de Fátima Queiroz Vieira, Ph.D.
Examinador

Campina Grande, Paraíba, Brasil, Junho de 2010

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

R119a Rached, Taciana Saad.
Arcabouço de software para auxiliar no desenvolvimento de sistemas interface cérebro-computador pervasivos /Taciana Saad Rached. – Campina Grande, 2010.
71 f.: il. col.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.
Orientador: Prof. Dr. Angelo Perkusich.
Referências.

1. Interface Cérebro-Computador. 2. Sistema ICC Pervasivo. 3. Deficiência Motora Severa. I. Título.

CDU 004.5(043)

ARCABOUÇO DE SOFTWARE PARA AUXILIAR NO DESENVOLVIMENTO DE SISTEMAS INTERFACE CÉREBRO-COMPUTADOR PERVASIVOS

TACIANA SAAD RACHED

Dissertação Aprovada em 14.06.2010

ANGELO PERKUSICH, D.Sc., UFCG
Orientador

MARIA DE FÁTIMA QUEIROZ VIEIRA, Ph.D., UFCG
Componente da Banca

HYGGO OLIVEIRA DE ALMEIDA, D.Sc., UFCG
Componente da Banca

CAMPINA GRANDE - PB
JUNHO - 2010

Agradecimentos

Agradeço aqueles que tiveram participação essencial para a finalização deste trabalho. Em primeiro lugar, a Deus por ter me inspirado com força e coragem, sabedoria e serenidade para melhor desenvolver este projeto. Quero lembrar ainda e agradecer a Capes que me apoiou na realização da minha pesquisa e me deu o suporte financeiro necessário. De forma particular, agradeço aos meus professores Angelo Perkusich, Hyggo Almeida e Marcos Morais pela orientação, dedicação, compreensão, amizade e apoio, ajudando-me a aprofundar meu conhecimento acerca do tema escolhido. Agradeço aos funcionários do Laboratório de Sistemas Embarcados e Computação Pervasiva: Aleixo, Simões e Luis Carlos e aos funcionários da COPELE: Ângela e Pedro pela amizade e pela disposição em ajudar. Agradeço aos amigos: Kalina, Dielle, Mateus, Leo e Diego pelo carinho e incentivo. E, em especial, agradeço aos meus pais e meus irmãos que sempre acreditaram no meu potencial e na minha dedicação aos trabalhos acadêmicos e ainda estiveram ao meu lado ajudando-me a enfrentar as dificuldades e fazendo-me perseverar até o término desta pesquisa. Ainda o meu muito obrigado a todos que contribuíram direta ou indiretamente para a realização deste trabalho.

Resumo

O desenvolvimento de tecnologias assistivas é muito importante no que diz respeito à vida de pessoas portadoras de algum tipo de deficiência. Na implementação dessas tecnologias o objetivo é fornecer ou estender habilidades funcionais para pessoas com necessidades especiais. As tecnologias assistivas promovem uma vida mais independente, inclusão social e melhora na qualidade de vida das pessoas citadas.

Apesar dos esforços despendidos para promover a inserção de pessoas deficientes na sociedade, ainda são limitados os recursos disponíveis para habilitar a comunicação entre pessoas portadoras de deficiência motora severa e o mundo. Neste contexto, a Interface-Cérebro Computador (ICC) é uma tecnologia recente que permite a comunicação direta entre o cérebro de uma pessoa e os dispositivos disponibilizados no ambiente no qual ela está inserida. Sistemas ICC habilitam pessoas portadoras de deficiência motora severa a controlar os dispositivos citados através de suas atividades cerebrais.

ICC é uma tecnologia recente, mas bastante promissora. Apesar disso, ainda existem algumas dificuldades encontradas para que os sistemas ICC possam ser aplicados no dia-a-dia de pessoas portadoras de deficiência motora severa. Alguns dos desafios encontrados incluem a precisão, velocidade e mobilidade do sistema. A maioria parte dos sistemas ICC são aplicações para *desktop*, sendo principalmente empregados em pesquisas clínicas.

O desenvolvimento de sistemas que integram ICC, dispositivos móveis e serviços pervasivos, além de habilitar pessoas portadoras de deficiência motora severa a controlar dispositivos, também fornece mobilidade a essas pessoas, permitindo que a interação entre elas e os dispositivos inseridos em seus meios possa ser realizada em qualquer lugar e a qualquer hora.

Neste trabalho é introduzida uma arquitetura para o desenvolvimento de sistemas ICC pervasivos para o controle de dispositivos multimídia. Além disso, foi desenvolvido um arcabouço de *software* para auxiliar no desenvolvimento de aplicações ICC multimídia. O usuário necessita de um equipamento de eletroencefalografia (EEG) para aquisição dos sinais cerebrais, um dispositivo móvel para o processamento, execução do arcabouço de *software* e personalização de serviços a partir do perfil do usuário, para interagir com qualquer dispositivo multimídia. Também foi implementado um sistema ICC pervasivo para validar a arquitetura introduzida. No desenvolvimento do sistema, foi implementado um equipamento de EEG, o algoritmo de classificação do sinal cerebral e uma aplicação multimídia para o controle do centro de mídias XBMC. O sistema ICC pervasivo desenvolvido no contexto deste trabalho permite que usuários executem conteúdo multimídia no XBMC. Para obtenção de resultados com relação à usabilidade do sistema, foi planejado e elaborado o ensaio de usabilidade para avaliação do sistema.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

O desenvolvimento de tecnologias assistivas vem sendo alvo de diversos estudos devido ao grande número de pessoas portadoras de necessidades especiais. O objetivo nessas pesquisas é desenvolver ou estender habilidades funcionais para os indivíduos que possuem alguma deficiência, promovendo vida independente, inclusão social e melhora da qualidade de vida [22]. Essas tecnologias incluem brinquedos e roupas adaptadas, computadores, *software* e *hardware* especiais, que contemplam questões de acessibilidade, dispositivos para adequação da postura sentada, recursos para mobilidade manual e elétrica, equipamentos de comunicação alternativa, chaves e acionadores especiais, aparelhos de escuta assistida, auxílios visuais e materiais protéticos, entre outros. Exemplos de tecnologias assistivas são apresentados na Figura 1.1.

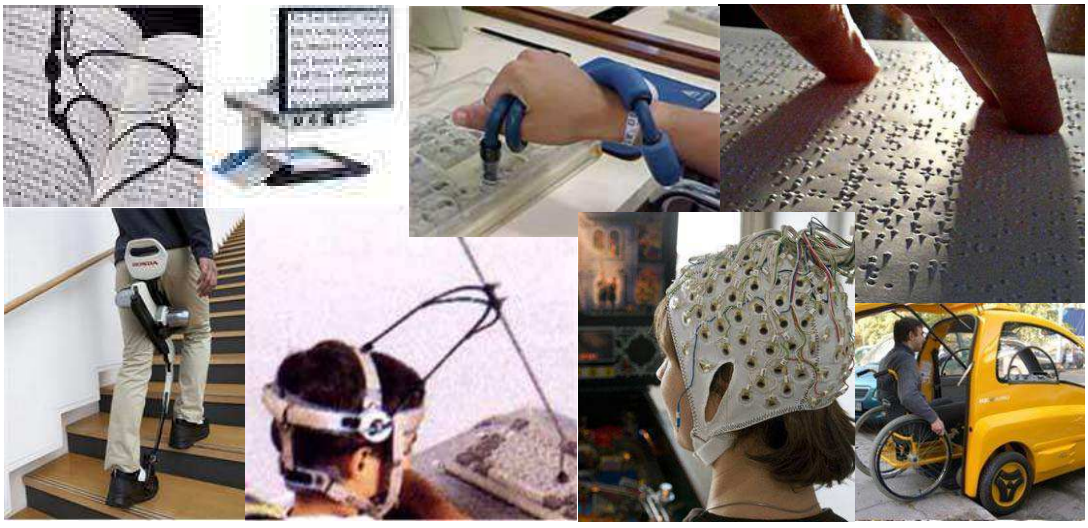

Figura 1.1: Exemplos de Tecnologias Assistivas

Neste contexto, Interação Homem-Máquina (IHM) é a área de pesquisa em crescimento no que diz respeito ao desenvolvimento de interfaces inovadoras e mais ergonômicas, para facilitar a interação entre indivíduos com necessidades especiais e o ambiente no qual eles estão inseridos [35]. Exemplos dessas interfaces incluem voz [38], visão [43], e outros dispositivos

de entrada/saída em realidade virtual [15].

Apesar dos esforços para o desenvolvimento de novas tecnologias para auxiliar pessoas com algum tipo de deficiência, o grupo de pessoas que possuem sérios problemas motores, foco no contexto deste trabalho, ainda é pouco favorecido por grande parte das soluções existentes. Esse grupo é composto por milhões de pessoas no mundo [13] que possuem doenças neuromusculares como esclerose lateral amiotrófica [1], esclerose múltipla [8], lesão medular [11] e paralisia cerebral [4]. Essas doenças, geralmente, causam paralisia e perda da fala, mas não afetam o sistema cognitivo. Como conseqüência, os mecanismos de comunicação entre pessoas com deficiência motora severa e os dispositivos encontrados no ambiente no qual elas estão inseridas são limitados.

Uma proposta de solução emergente e promissora que vem sendo foco de diversas pesquisas para ajudar na reabilitação de pessoas portadoras de deficiência motora severa é um novo canal de comunicação entre o cérebro de uma pessoa e o ambiente externo, chamada Interface Cérebro-Computador (ICC) [41]. ICC é tema de pesquisa em IHM [25] em estudos clínicos e desenvolvimento de dispositivos eletrônicos de consumo. Neste contexto, a recente disponibilidade de dispositivos eletrônicos de consumo para serem usados em aplicações ICC, como o fone *MindSet* da *NeuroSky* [9], é a principal tendência para permitir a implementação de sistemas ICC de baixo custo com o objetivo de promover a comunicação entre indivíduos com deficiência motora e o ambiente.

ICC são sistemas que habilitam o usuário a trocar informações com o ambiente e controlar dispositivos através da sua atividade cerebral, sem o uso dos músculos neurais como caminho de saída do cérebro [56]. Como discutido por Wang em [53], sistemas ICC são usados em diferentes tipos de situações, como: na reabilitação, neurociência, sistemas de monitoramento de atenção, psicologia cognitiva e na área psicomotora.

1.1 Motivação

Diante dos avanços nas pesquisas em IHM para criação de novos canais de comunicação para indivíduos portadores de deficiências, os sistemas ICC surgiram como uma alternativa para habilitar a interação entre pessoas portadoras de deficiência motora severa e os dispositivos disponibilizados no ambiente no qual estão inseridas. Apesar de permitir a comunicação e interação direta entre o cérebro e o ambiente, ainda existem desafios para que os sistemas ICC possam ser usados no dia-a-dia de pessoas portadoras de deficiência motora severa. Neste contexto, um problema importante a ser tratado é a mobilidade dos usuários de sistemas ICC. A maioria dos estudos em ICC é voltada para o desenvolvimento de sistemas baseados em desktop, tendo ainda sua principal aplicação em pesquisas clínicas. Como conseqüência, usuários de sistemas ICC possuem mobilidade limitada, isolamento social e dificuldades para realizar

tarefas simples do dia-a-dia. Esses problemas têm impacto fisiológico, psicológico e social na vida de pessoas com deficiência.

Nos últimos anos, diversas empresas vêm desenvolvendo equipamentos de eletrônica de consumo de baixo custo para aquisição dos sinais cerebrais. Esses dispositivos são fáceis de usar e podem ser usados em qualquer ambiente. Exemplos desses equipamentos incluem o *MindSet* desenvolvido pela *NeuroSky* [45] e o *EPOC headset* [57] fabricado pela *Emotiv*. Além da fabricação desses dispositivos, algumas empresas vêm produzindo chips que realizam a aquisição e grande parte do processamento do sinal. Esses chips custam em média um dólar o que torna atrativo para universidades e empresas desenvolverem seus próprios equipamentos de aquisição dos sinais cerebrais.

O advento de tecnologias sem fio e o crescimento no número de dispositivos portáteis inteligentes como PDAs e internet *tablets* tornaram possível o desenvolvimento de sistemas onde mobilidade é uma realidade. Além disso, é importante ressaltar que a intervenção do usuário deve ser a mínima possível para que tarefas simples possam ser realizadas com muito pouca ou nenhuma intervenção humana. Neste contexto, os conceitos e técnicas de Computação Pervasiva [55] são muito importantes, pois, além da mobilidade, a computação nômade e ubíqua é aplicada considerando mínima a intervenção humana. A Computação Pervasiva visa integrar os ambientes virtual e real para criar uma computação invisível aos olhos dos usuários, onde eles interagem com o sistema de computação pervasiva de maneira integrada.

A integração de sistemas ICC com dispositivos inteligentes e serviços pervasivos provê uma nova perspectiva de vida para pessoas com deficiência. Aplicações ICC pervasivas habilitam usuários a desempenhar diversas atividades através de seus sinais cerebrais e dispositivos inteligentes, como: controle da temperatura de um ambiente, envio de mensagens para um amigo, mudança de canal de uma televisão, entre outras.

Possibilitar que pessoas portadoras de deficiência motora severa possam interagir com dispositivos presentes no ambiente no qual elas estão inseridas a qualquer hora e em qualquer lugar com o uso de sistemas ICC, dispositivos inteligentes e serviços pervasivos motivou o desenvolvimento deste trabalho.

1.2 Objetivos

Neste trabalho tem-se como objetivo introduzir uma arquitetura e um arcabouço de software para auxiliar no desenvolvimento de sistemas ICC pervasivos para o controle de dispositivos multimídia.

Para atingir o objetivo final no trabalho, alguns objetivos específicos devem ser alcançados:

- Desenvolvimento de um equipamento de aquisição e processamento dos sinais cerebrais.

- Realização de treinamento.
- Definição do modelo da arquitetura a ser introduzida neste trabalho.
- Implementação do arcabouço de componentes de software.
- Definição e implementação do estudo de caso.
- Definição dos testes de usabilidade.

1.3 Metodologia

Para o desenvolvimento deste trabalho as seguintes etapas foram definidas:

- Desenvolvimento de um equipamento de quatro canais para leitura dos sinais a partir de eletrodos de Ag/AgCl que devem ser aplicados a superfície do escalpo do usuário. Apesar de não ser o foco deste trabalho, escolheu-se desenvolver este equipamento, devido a disponibilidade de tecnologias de baixo custo que implementam algoritmos de pré-processamento e seleção das características dos sinais cerebrais. No Capítulo 3, é realizada uma discussão sobre a motivação para o desenvolvimento do *hardware* para aquisição da atividade e os componentes usados. No Capítulo 4, são abordados a implementação e o funcionamento do equipamento.
- Realização de treinamento com os usuários do sistema. Em um sistema ICC é importante realizar uma etapa de treinamento para que o usuário aprenda a operar o sistema. No Capítulo 3, é definido o procedimento a ser realizado durante o treinamento. No Capítulo 4, é discutida a realização do treinamento.
- Definição de um modelo da arquitetura para auxiliar no desenvolvimento de sistemas ICC pervasivos para o controle de dispositivos multimídia. A maioria dos sistemas ICC disponíveis atualmente são aplicações para desktop, conseqüentemente o usuário desse tipo de sistema possui mobilidade limitada. Na definição de um modelo de arquitetura para sistemas ICC pervasivos tem-se o objetivo de incentivar o desenvolvimento deste tipo de aplicação para que esses sistemas possam ser usados em qualquer lugar e a qualquer hora por pessoas portadoras de deficiência motora severa. A arquitetura é abordada no Capítulo 3.
- Implementação de um arcabouço de componentes de software para auxiliar desenvolvedores de aplicações multimídia controladas por sistemas ICC. Esse arcabouço deve ser multiplataforma e permitir a adição de novos componentes sem que o desenvolvedor tenha a preocupação com a comunicação entre o arcabouço e o restante do sistema. No

desenvolvimento deste arcabouço tem-se o objetivo de facilitar e motivar o desenvolvimento de aplicações ICC pervasivas. A arquitetura e o funcionamento do arcabouço são apresentados no Capítulo 3. A implementação do arcabouço é abordada no Capítulo 4.

- Desenvolvimento de uma aplicação para validar a arquitetura e o arcabouço desenvolvidos neste trabalho. O estudo de caso trata-se do uso do sistema ICC pervasivo desenvolvido neste trabalho no controle do centro de mídias XBMC. Nessa aplicação, o usuário manipula conteúdo multimídia a partir de sua atividade cerebral. No Capítulo 4, é apresentada a aplicação.

- Definição dos testes de usabilidade a serem realizados para avaliação do sistema ICC pervasivo desenvolvido no contexto deste trabalho. O objetivo na realização dos ensaios de usabilidade é detectar falhas no *software* e no *hardware*, e identificar as principais dificuldades encontradas pelo usuário no uso do sistema. Nos Capítulos 3 e 4, são discutidos o planejamento e a elaboração dos testes de usabilidade.

1.4 Organização do Texto

O restante desse texto está organizado em quatro capítulos. No Capítulo 2 são apresentados os conceitos que embasaram o desenvolvimento deste trabalho. No Capítulo 3 são discutidas as escolhas realizadas para o desenvolvimento deste trabalho. No Capítulo 4 é apresentado o desenvolvimento deste trabalho. Por fim, no Capítulo 5 são mostradas as conclusões e os trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo, é abordada a Interface Cérebro-Computador, sendo discutidas as etapas necessárias para o desenvolvimento de um sistema ICC. São apresentados os tipos de sistemas ICC de acordo com o método de aquisição do sinal cerebral, os tipos de características que compõem um sinal de EEG, os canais de aquisição, o processamento do sinal, a etapa de treinamento necessária para que o usuário de um sistema ICC aprenda a controlá-lo e, por fim, algumas aplicações relacionadas com este trabalho. Este trabalho foi desenvolvido baseado nos conceitos discutidos neste Capítulo.

2.1 Interface Cérebro-Computador

As Interfaces Cérebro-Computador são sistemas que habilitam indivíduos a trocar informações com o ambiente e controlar dispositivos através da sua atividade cerebral, ou seja, mesmo sem o uso dos nervos periféricos e músculos como caminho de saída do cérebro é possível estabelecer uma comunicação entre esses indivíduos e o mundo [56]. Exemplos de dispositivos que podem ser controlados por sistemas ICC são cadeira de rodas, robôs ou computadores. Na Figura 2.1 é apresentado o diagrama de blocos de um sistema ICC básico.

2.1.1 Aquisição dos Sinais

As tecnologias de aquisição dos sinais cerebrais disponíveis atualmente podem ser divididas em dois grupos: métodos invasivos ou não-invasivos. Os métodos invasivos consistem da aplicação de eletrodos diretamente no cérebro, ou seja, é necessária uma intervenção cirúrgica para o uso desses métodos. Na Figura 2.2 é apresentado um exemplo de método invasivo. Os métodos não-invasivos permitem a aquisição dos sinais cerebrais sem intervenção cirúrgica. Esses métodos podem ser usados para realização de testes em humanos sem comprometer a saúde destes [48]. Na Figura 2.3 é apresentado um exemplo de método não-invasivo.

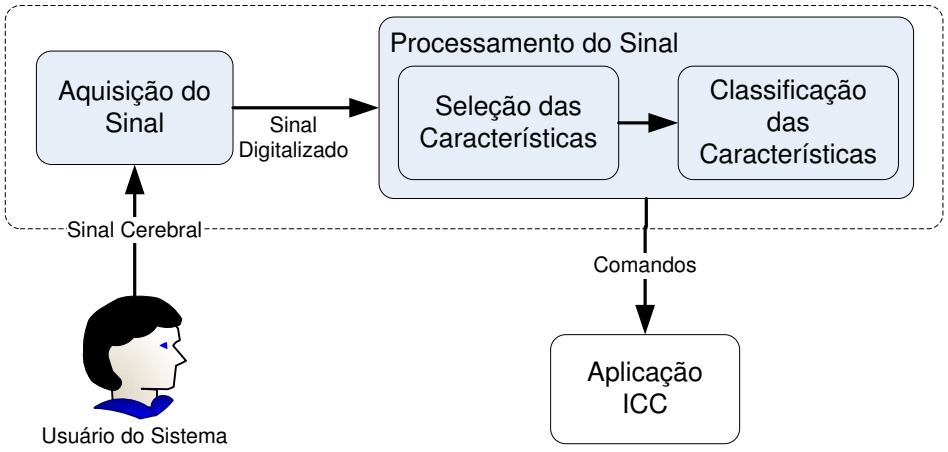

Figura 2.1: Diagrama de blocos de um sistema ICC básico

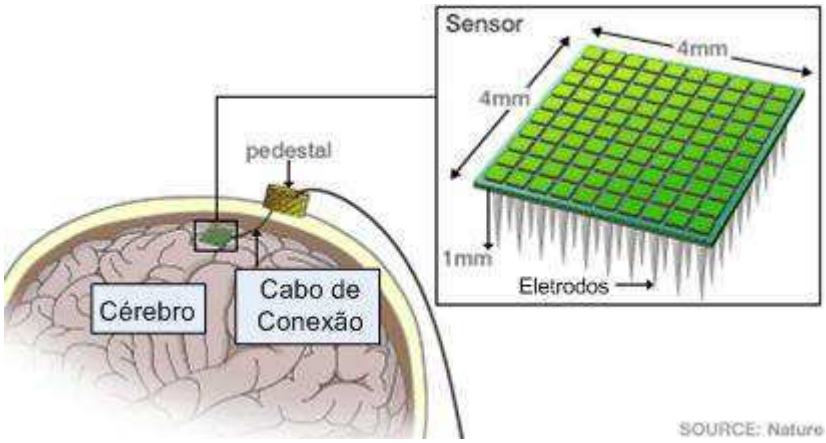

Figura 2.2: Representação de método invasivo usado para aquisição dos sinais cerebrais

Existem diversos métodos não-invasivos de monitoramento dos sinais cerebrais. Esses métodos incluem a Magnetoencefalografia (MEG) [30], a Ressonância Magnética [37], a Espectroscopia de Infravermelho Próximo [23] e a Eletroencefalografia (EEG), entre outros. EEG é usado na aquisição da atividade bioelétrica do cérebro por meio de um conjunto de até 256 eletrodos de superfície aplicados ao couro cabeludo do usuário. As flutuações de tensão detectadas pelo EEG são somatórios da atividade elétrica de grandes populações de neurônios corticais. EEG é a técnica mais usada, em relação às citadas anteriormente, em sistemas ICC devido a sua portabilidade, ou seja, o EEG pode ser usado em qualquer ambiente, facilidade de uso, baixo custo e alta resolução temporal [33].

Sistemas ICC que utilizam sinais de EEG podem ser subdivididos em diversos grupos de acordo com o tipo de sinais eletrofisiológicos usado. Esses sinais eletrofisiológicos incluem os potenciais evocados visuais (PEV), os potenciais evocados visuais no estado estável (PEVEE), os potencias evocados P300, os potenciais corticais lentos e as características rítmicas delta, teta, alfa, beta e gama. Na Tabela 2.1 são apresentadas as características rítmicas e os diferentes

Figura 2.3: Foto de equipamento usado para adquirir os sinais cerebrais não invasivamente

tipos de atividades que tendem a gerar essas características.

Tipo de Característica	Variação de Freqüência	Estados e Condições Mentais
Delta	0,1 a 3,99 Hz	Associadas ao sono profundo
Teta	4 a 7,99 Hz	Estado sonolento
Alfa	8 a 12,99 Hz	Relaxado, mas não sonolento, consciente
Beta	13 a 29,99 Hz	Focado, alerta
Gama	30 a 50 Hz	Associado com pico de concentração

Tabela 2.1: Freqüências cerebrais mais conhecidas e atividades que tendem a gerá-las

Sistemas ICC que dependem da utilização da atividade voluntária de algum músculo ou dos nervos periféricos para interação entre o cérebro de um indivíduo e o ambiente são classificados como dependentes [56]. Em ICC, sistemas que utilizam PEV são classificados como dependentes, pois a geração destes sinais depende do controle do nervo oculomotor da direção do olhar, ou seja, são utilizados os caminhos normais, nervos periféricos e músculos, para geração da atividade cerebral. Potenciais visuais evocados são obtidos como resposta do cérebro à estimulação visual em uma determinada frequência [54]. No caso em que a freqüência de repetição de estímulos é maior que 6 Hz, os potenciais visuais evocados atingem o estado estável [42].

Müller-Putz e Pfurtscheller [40] usaram um sistema ICC baseado nos sinais PEVEE para controlar uma prótese de mão. Foram usadas quatro luzes vermelhas localizadas em posições específicas na prótese. Para estimular seus cérebros a gerar sinais do tipo PEVEE, os participantes foram instruídos a focar na cintilação das luzes localizadas na prótese, por pelo menos um segundo, para que a prótese realizasse a correspondente ação. O movimento para a direita, por exemplo, correspondia a uma estimulação visual de freqüência 7 Hz. Nesse trabalho, os participantes realizaram o controle da prótese de modo que ela se movimentasse para a direita e

para a esquerda. Além disso, a prótese também foi controlada para desempenhar os movimentos de abertura e fechamento dos dedos.

Sistemas ICC são definidos como independentes quando não é necessária nenhuma atividade muscular voluntária, ou seja, quando é realizada uma comunicação direta entre o cérebro e o computador [56]. Potenciais evocados P300 [24] são exemplos de sinais usados em sistemas ICC independentes. A componente P300 de eventos potenciais relacionados (EPR) é um pico positivo que acontece 300 milissegundos após um estímulo. Os EPR consistem de uma sequência de flutuações de tensões positivas e negativas. A componente P300 é obtida a partir da distinção entre dois estímulos. Esse tipo de procedimento é chamado "paradigma *oddbal*" e trata-se da apresentação de uma série aleatória de dois estímulos onde um deles ocorre com pouca freqüência (o *"oddbal"*). Para gerar a componente P300, o usuário deve notar a ocorrência do estímulo *"oddbal"*, ignorando o estímulo que ocorre freqüentemente [27].

Edlinger et al. [26] apresentaram um sistema ICC para aplicações em realidade virtual. Neste trabalho foram usados os potenciais evocados P300 para permitir que indivíduos controlassem uma residência virtual inteligente. Nesse experimento os usuários puderam ligar/desligar a lâmpada, abrir/fechar as portas e janelas, mudar canal e volume da televisão, usar o telefone, tocar uma música, operar as câmeras da entrada, caminhar ao redor da casa e se mover para uma localização específica da casa inteligente. Nesse trabalho foram desenvolvidas máscaras especiais de controle para todos os diferentes comandos necessários. Os símbolos referentes a cada máscara foram apresentados em uma tela. Para realizar uma das tarefas citadas anteriormente, os participantes tiveram que se concentrar nos símbolos destacados aleatoriamente.

Os Potenciais corticais lentos [36] são mudanças de tensões lentas, geradas no córtex, que ocorrem entre 0,5 e 10 segundos. PCL negativos são associados com movimentos e outras funções envolvendo a ativação cortical. Potenciais corticais lentos positivos são associados com a redução da ativação cortical.

Birbaumer et al. [16, 17] e Hinterberger et al. [28, 29] mostraram um dispositivo de comunicação para usuários com deficiência motora grave chamado *though translation device* (TTD). O TTD é um dispositivo, com realimentação visual ou auditiva, que permite ao usuário usar os potenciais corticais lentos para soletrar palavras e escrever mensagens. O TTD consiste de um dispositivo de treinamento e um programa de suporte a linguagem. Durante a fase de treinamento os indivíduos aprenderam como controlar seus potenciais corticais lentos. Durante a fase de soletração os participantes selecionaram letras com seus potenciais corticais lentos. No programa de suporte a linguagem o alfabeto primeiramente (nível 1) foi separado em bancos de letras que foram apresentados sucessivamente na parte inferior da tela por alguns segundos. A seleção de um dos bancos de letras foi realizada pelos usuários através de uma mudança nos potenciais corticais lentos. Após ser selecionado, o banco de letras foi dividido em dois novos bancos de letras. Esse procedimento foi repetido até que existissem apenas dois bancos de letras

com apenas uma letra em cada um deles. A última letra selecionada foi mostrada no campo de texto na parte superior da tela e uma nova etapa de seleção foi iniciada.

Características do sinal podem ser extraídas de diversas bandas de freqüência do sinal de EEG adquirido. McFarland et al. em [39] apresentaram um sistema ICC baseado na atividade rítmica alfa. Esse sistema foi desenvolvido para permitir ao usuário controlar um cursor na tela de um computador. Para que o participante pudesse utilizar esse sistema, foi necessária a realização de uma etapa de treinamento para que ele aprendesse a controlar as componentes alfa responsáveis pelo movimento do cursor.

Posicionamento dos Eletrodos e Nomenclatura

Os sinais de EEG podem ser medidos a partir de diferentes áreas do cérebro. A seleção dos canais de monitoramento dos sinais de EEG depende das características do sinal que serão usadas pelo sistema. O cérebro é composto por quatro regiões: o lóbulo Frontal, o lóbulo Parietal, o lóbulo Occipital e o lóbulo Temporal. Cada lóbulo desempenha diferentes funções. O lóbulo frontal está relacionado com a elaboração do pensamento, planejamento, programação das necessidades individuais e emoção. O lóbulo parietal é responsável pela sensação de dor, tato, gustação, temperatura e pressão. O lóbulo temporal está relacionado com o sentido da audição, possibilitando o reconhecimento de tons específicos e intensidade do som. O lóbulo occipital é responsável pelo processamento da informação visual. O sulco central separa os lóbulos Frontal e Parietal. As áreas situadas adiante do sulco central relacionam-se com a motricidade, enquanto as situadas atrás deste sulco relacionam-se com a sensibilidade. As estruturas cerebrais podem ser observadas na Figura 2.4.

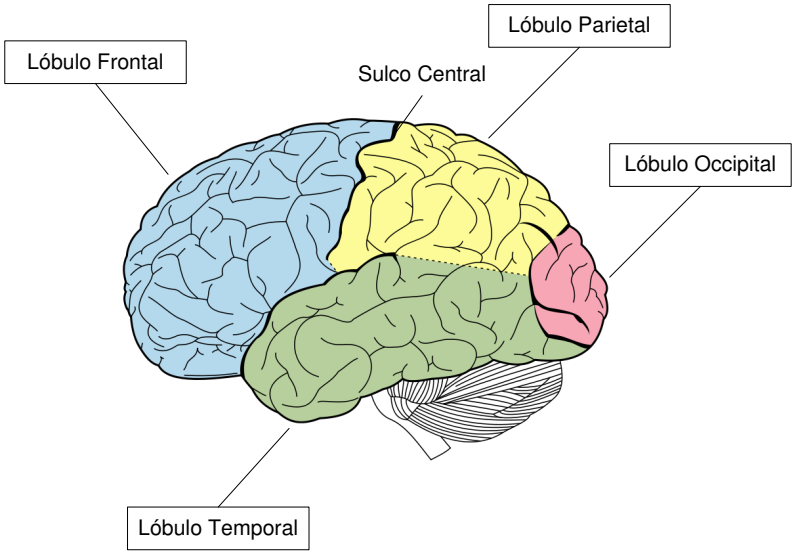

Figura 2.4: Representação das estruturas anatômicas do cérebro

Os eletrodos são aplicados na superfície do escalpo do usuário de acordo com o sistema

Internacional 10-20 para a aquisição dos sinais de EEG. O sistema Internacional 10-20 é usado para padronizar a localização dos eletrodos com o objetivo de facilitar a realização de comparações entre diferentes estudos. Nesse sistema os nomes dos eletrodos são compostos por algumas letras e um número. As letras se referem às estruturas anatômicas (lóbulos Frontal, Parietal, Occipital e Temporal e o sulco central). Os números denotam a linha sagital (anterior ou posterior). Os números ímpares correspondem ao hemisfério esquerdo, enquanto os números pares correspondem ao direito. A letra "z" minúscula é usada em nomes de eletrodos localizados na linha sagital central [34]. O sistema internacional 10-20 é apresentado na Figura 2.5.

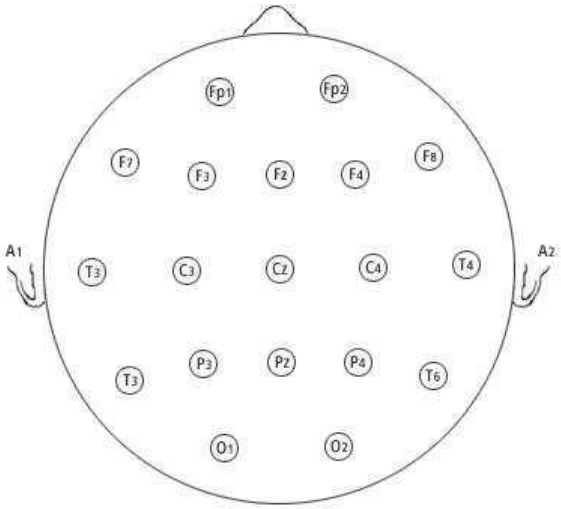

Figura 2.5: Sistema Internacional 10-20

2.1.2 Treinamento

Em um sistema ICC é necessária a realização de um mínimo de treinamento para que o usuário possa operá-lo. O treinamento serve para ensinar uma máquina e ajustar os parâmetros do modelo para melhor atender o usuário e as propriedades do sinal cerebral. Durante o treinamento são adquiridos sinais de EEG enquanto os participantes desempenham certa tarefa. Exemplos dessas atividades incluem execução ou imaginação dos movimentos dos braços direito e esquerdo. Os algoritmos de decodificação são individualmente adaptados para os usuários que desempenham a tarefa previamente definida. A partir dos sinais adquiridos durante o treinamento os algoritmos de aprendizado podem inferir estruturas estatísticas do respectivo estado cerebral. Portanto, é necessário que o indivíduo produza repetidamente certo estado cerebral durante uma sessão de calibração. Mesmo em uma pequena quantidade de dados, as máquinas de aprendizado atuais podem extrair modelos espaço-temporal destes estados cerebrais, os quais são facilmente reutilizáveis nas sessões de realimentação subseqüentes.

2.1.3 Processamento do sinal

Em sistemas ICC o processamento do sinal de EEG é dividido em duas etapas: a seleção das características e a classificação.

Extração das Características

Na seleção das características tem-se o objetivo de medir as características do sinal que codificam a saída [44]. As características do sinal são apresentadas no domínio do tempo (potenciais evocados P300) ou no domínio da freqüência (ritmos alfa e beta).

A seleção das características do sinal no domínio do tempo é realizada com o uso de métodos como a análise de componentes independentes e a transformada rápida de Fourier, entre outros. No domínio da freqüência são usados métodos baseados no tempo, no espaço e na combinação entre tempo e espaço. Métodos baseados no tempo incluem a filtragem passa-faixa, análise espectral baseada na transformada de Fourier, métodos paramétricos e o uso de *wavelets*. Métodos baseados no espaço incluem filtros de Laplace, análise de componentes principais e análise de componentes independentes. Métodos baseados na combinação entre tempo e espaço incluem análise de componentes no tempo e espaço e modelos auto-regressivos multivariados.

Classificação

A classificação do sinal trata-se do mapeamento das características do sinal em comandos para o dispositivo de saída usando algoritmos de tradução. Algoritmos de tradução podem ser simples ou complexos. Algoritmos simples incluem as equações lineares. Os métodos complexos incluem as redes neurais e as máquinas de suporte vetorial [44].

Uma Máquina de Suporte Vetorial (MSV) [32] é uma técnica de aprendizado de máquina baseada na estatística e fundamentada nos princípios de Minimização do Risco Estrutural e na teoria da dimensão *Vapnik-Chervonenkis* (VC). Basicamente uma Máquina de Suporte Vetorial é uma máquina linear cuja idéia principal consiste em construir um hiperplano como superfície de decisão onde a margem de separação entre os exemplos rotulados como positivos e negativos tem de ser máxima. Uma noção central à construção do algoritmo de aprendizado por vetor de suporte é o núcleo do produto interno entre um vetor de suporte e o vetor retirado do espaço de entrada. Os vetores de suporte consistem em um pequeno subconjunto de treinamento extraído pelo algoritmo. Dependendo de como este núcleo é gerado, podemos construir diferentes máquinas de aprendizado caracterizadas por superfície de decisão própria.

2.1.4 Realimentação

A realimentação é um procedimento que busca permitir ao usuário adquirir o controle sobre um parâmetro autônomo. Os participantes recebem informações visuais, auditivas ou táteis sobre

sua atividade cardiovascular, temperatura, condutividade da pele, atividade muscular, atividade elétrica cerebral, ou nível de oxigênio no sangue. Nos ensaios discretos ou contínuos, tarefas são apresentadas aos participantes com o objetivo de aumentar ou diminuir a atividade de interesse. Por meio do sinal de realimentação, os participantes recebem informações contínuas sobre a alteração da atividade. Ao final do experimento, os participantes são informados sobre o seu desempenho. Se os participantes são repetidamente treinados, eles aprendem a manipular a atividade de interesse, ao menos em certa medida, sob controle voluntário ou consciente [36].

2.1.5 Aplicações

A aplicação de sistemas ICC no dia-a-dia depende da resolução de alguns problemas como velocidade, precisão e mobilidade, entre outros. A maior parte dos sistemas ICC é usada para realizar o controle básico de um ambiente, como ligar/desligar uma lâmpada, mudar o canal de uma televisão ou alterar a temperatura de um ambiente. Além disso, sistemas ICC são usados no controle de cadeira de rodas, robôs e em realidade virtual. Nesta Seção são discutidos os estudos mais relevantes no contexto deste trabalho.

No sistema Berlin brain-computer interface (BBCI) [34] o processamento das tarefas é dividido em diferentes computadores devido ao grande número de informações para ser processada. Esses computadores se comunicam através de interfaces cliente-servidor. Na Figura 2.6 é apresentado o sistema distribuído BBCI.

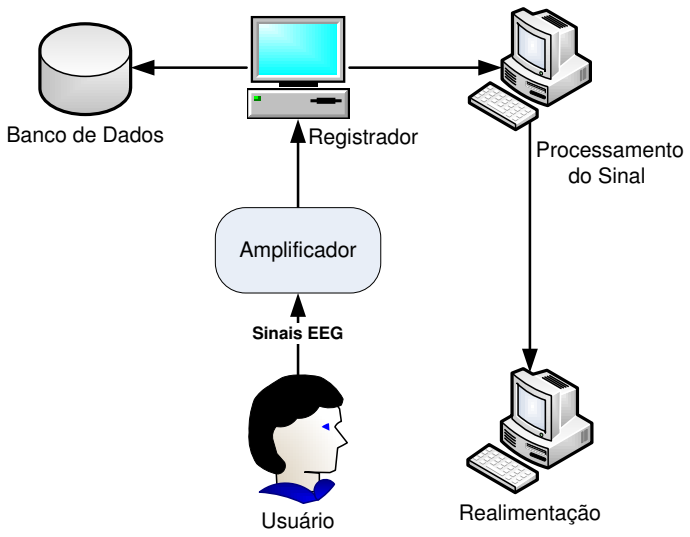

Figura 2.6: Sistema distribuído usado em BBCI

Neste sistema a aquisição dos sinais de EEG é realizada a partir de 128 eletrodos colocados na superfície da cabeça do usuário. Esses sinais são amplificados, digitalizados e enviados a um computador através de cabos de fibra ótica. Nesse computador são realizadas algumas operações simples de pré-processamento e o armazenamento de dados. Esse computador também

atua como um servidor remoto de acesso de dados o qual suporta conexões de até dez clientes. Um segundo computador realiza as etapas de processamento: seleção e classificação das características do sinal. Na etapa de classificação o sinal é traduzido em um sinal de comando. Esse computador funciona como um servidor para várias aplicações multimídia usadas para realimentação. A aplicação multimídia é executada em um terceiro computador.

A utilização de um grande número de eletrodos para monitoramento dos sinais de EEG dificulta o manuseio do equipamento de aquisição pelo usuário. O uso de cabos na transmissão dos sinais adquiridos e o emprego de vários computadores na realização do processamento de diferentes tarefas restringem a mobilidade do indivíduo e dificultam o uso do sistema BBCI no dia-a-dia das pessoas.

Ebrahimi et al. [25] discutiram sobre o uso de sistemas interface cérebro-computador (ICC) como um novo canal de comunicação entre pessoas e aplicações multimídia. Nesse trabalho foi apresentada uma arquitetura de um sistema ICC para ajudar em pesquisas e no desenvolvimento de sistemas ICC para aplicações multimídia. Foi proposto um sistema distribuído no qual cada componente fornece serviços específicos para os outros componentes. A arquitetura introduzida nesse trabalho é mostrada na Figura 2.7.

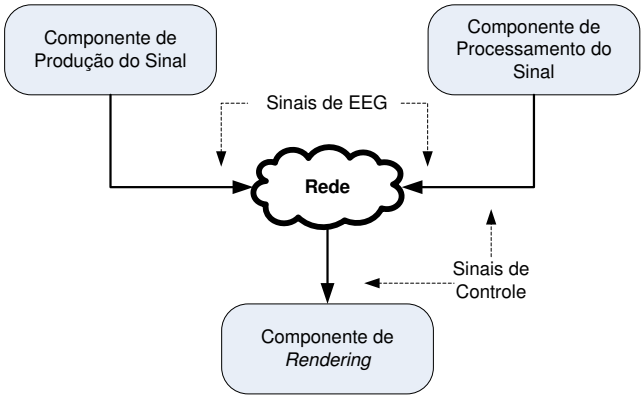

Figura 2.7: Arquitetura para auxiliar no desenvolvimento de sistemas ICC para o controle de dispositivos multimídia

Três tipos de componentes podem ser observados:

- Um componente de produção do sinal que é responsável por digitalizar os sinais de EEG adquiridos e transmitir eles para a unidade de processamento;
- Um componente de processamento do sinal que é encarregado de realizar as operações de pré-processamento, seleção das características e classificação;
- Um componente de *rendering* que é usado para realimentar o usuário visual ou auditivamente.

Os algoritmos usados no desenvolvimento dos componentes foram implementados Java, C e Matlab.

Sambasivan and Jackson em [47] apresentaram um protótipo de um sistema ICC pervasivo. Nesse protótipo foi usada a tecnologia espectroscopia funcional por infravermelho próximo para medir o nível de oxigênio no sangue dependendo da atividade neural. Uma página principal com diversos ícones foi mostrada ao usuário em uma tela como a de uma televisão. A seleção dos ícones foi realizada em movimento circular no sentido horário. Os ícones foram selecionados quando o participante contava os números mentalmente. A página principal com os ícones, movimentos e direções é mostrada na figura 2.8.

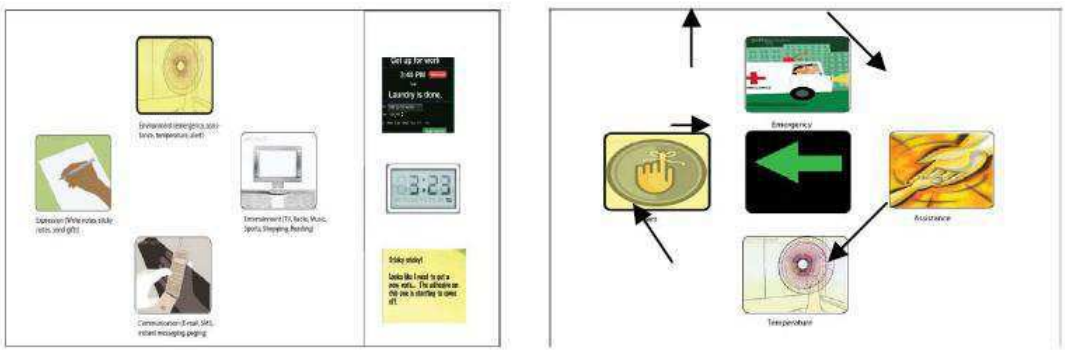

Figura 2.8: Na figura da esquerda é mostrada a página principal e a na da direita é ilustrado o ambiente de controle

A ciência de contexto foi um dos conceitos usados nesse trabalho. Um microfone e diversos sensores como os térmicos, de proximidade e de luminosidade foram embarcados no protótipo. Quando uma pessoa se aproximava do participante, por exemplo, automaticamente era iniciada a aplicação de conversação. O protótipo foi desenvolvido de acordo com os seguintes requisitos:

- O usuário deveria ter facilidade em aprender a usar o sistema;
- Flexibilidade, ou seja, tolerância a erros;
- Fácil de usar;
- O sistema deveria oferecer realimentação ao usuário;
- O sistema deveria permitir ao usuário reconhecer o estado atual;
- Quando o usuário estivesse em um estado indesejado, o sistema deveria permitir que o participante voltasse ao estado anterior;
- Aparência e funções consistentes.

O uso da tecnologia espectroscopia funcional por infravermelho próximo para monitorar a atividade cerebral em aplicações clínicas é muito comum, porém ainda não é viável aplicar esta tecnologia em atividades do dia-a-dia devido ao seu alto custo.

Teo et al. em [52] apresentaram uma interface de um *Media Communication Center* (MCC) controlada por um sistema ICC baseado nos potenciais evocados P300. A interface discutida neste trabalho possui funcionalidades integradas como um tocador de MP3, galeria de fotos, televisão e rádio, entre outras. Na figura 2.9 é ilustrada a interface gráfica do MCC apresentado nesse trabalho.

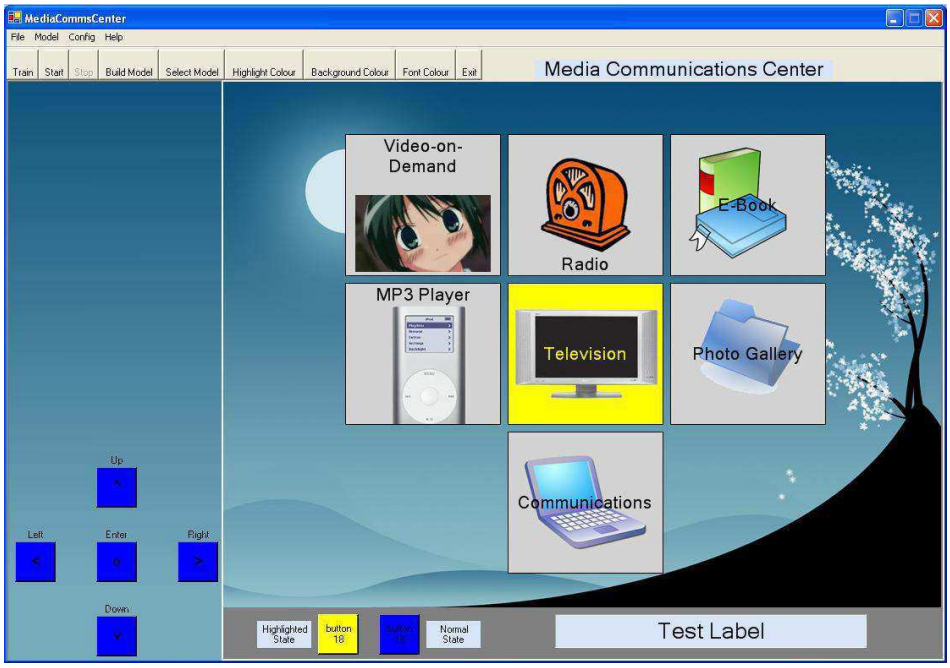

Figura 2.9: Interface Gráfica do MCC

O principal requisito usado na construção da interface mostrada na Figura 2.9 foi a facilidade de uso da interface. Esse requisito é importante em sistemas ICC para que o número de erros cometidos pelo usuário seja reduzido durante o uso do sistema, já que em sistemas ICC a velocidade dos sistemas ainda é um problema a ser resolvido.

Os trabalhos citados acima possuem como principal problema a mobilidade, pois são sistemas desenvolvidos para desktop.

2.2 Sumário

O desenvolvimento de sistemas Interface Cérebro-Computador vem sendo foco de diversas pesquisas recentemente, mas apesar disso existem diversos problemas a serem abordados para que sejam implementados sistemas cada vez mais eficientes. Esses problemas incluem a seleção do método de aquisição, dos canais usados para adquirir o sinal cerebral, do tipo de

característica, dos algoritmos usados na seleção e na classificação das características do sinal, do procedimento usado para treinar o usuário, do modo como a informação é apresentada ao usuário em uma interface, do tipo de realimentação, o contexto de uso do sistema, entre outros.

A escolha do método de aquisição, dos canais e do tipo de característica a serem usados pelo sistema são muito importantes para obtenção dos sinais que possuem maior relevância para aplicação em questão. Em um sistema desenvolvido para ser usado no dia-a-dia de uma pessoa não pode ser usado um equipamento de leitura do sinal cerebral grande, caro, e de difícil uso. Um canal de aquisição localizado na região temporal do cérebro, não é relevante para um sistema baseado em sinais classificados durante o fechamento/abertura dos olhos. Ou ainda, não tem sentido o uso da característica PEV em um sistema desenvolvido para pessoas que não possuem o controle dos músculos oculomotores. Na Seção 2.1.1, foram discutidos os métodos de aquisição, os tipos de características do sinal e os canais de aquisição.

A seleção dos algoritmos de processamento também possui grande impacto na eficiência de um sistema ICC. A escolha do algoritmo depende da característica do sinal a ser selecionada. Um algoritmo de seleção para características do sinal no domínio da freqüência não pode ser usado para características no domínio do tempo. Na seção 2.1.3, foram apresentadas as etapas de seleção e classificação das características do sinal cerebral.

Em um sistema ICC é importante treinar o usuário para que este possa operá-lo adequadamente. O procedimento a ser realizado durante o treinamento depende do tipo de característica do sinal que o usuário deve aprender a controlar e da aplicação a ser controlada pelo usuário. Em um sistema baseado nas características alfa, o usuário deve ser treinado para manter estas características em um determinado estado enquanto realiza uma determinada tarefa. O sistema desenvolvido neste trabalho é baseado nas características alfa e o procedimento usado durante o treinamento é discutido na Seção 3.4.

A disposição da informação em uma interface gráfica é bastante importante, pois essas informações são usadas para realimentar o usuário sobre seu estado cerebral atual e excitar sua atividade cerebral. O modo como os dados devem ser apresentados ao usuário está fora do escopo deste trabalho.

O contexto de uso do sistema é um dos principais desafios no desenvolvimento de um sistema ICC. A definição do contexto tem influência em todas as outras etapas discutidas anteriormente. Por exemplo, o sistema desenvolvido neste trabalho deve permitir ao usuário usá-lo em qualquer lugar e a qualquer hora. Conseqüentemente, o método de seleção deve ser fácil de usar, ser barato e poder ser usado em qualquer ambiente. O número de canais de aquisição deve ser reduzido para evitar uma grande quantidade de informação desnecessária para ser processada, pois esses dados devem ser processados por dispositivos móveis para fornecer uma maior mobilidade ao sistema. No Capítulo 4, é apresentado o sistema desenvolvido neste trabalho.

Capítulo 3

Arquitetura

Neste capítulo é introduzida uma arquitetura para o desenvolvimento de sistemas ICC usados no controle de dispositivos multimídia para serem utilizados por pessoas portadoras de deficiência motora severa [50, 51]. No desenvolvimento da arquitetura são considerados conceitos de mobilidade e pervasividade. Na integração entre sistemas ICC, dispositivos móveis e serviços pervasivos tem-se o objetivo de permitir que indivíduos com deficiência motora severa possam interagir com dispositivos existentes no ambiente no qual estão inseridos a qualquer hora. Como pode ser observado na Figura 3.1, a arquitetura é composta por dois módulos principais: o módulo ICC e o módulo Mapeamento.

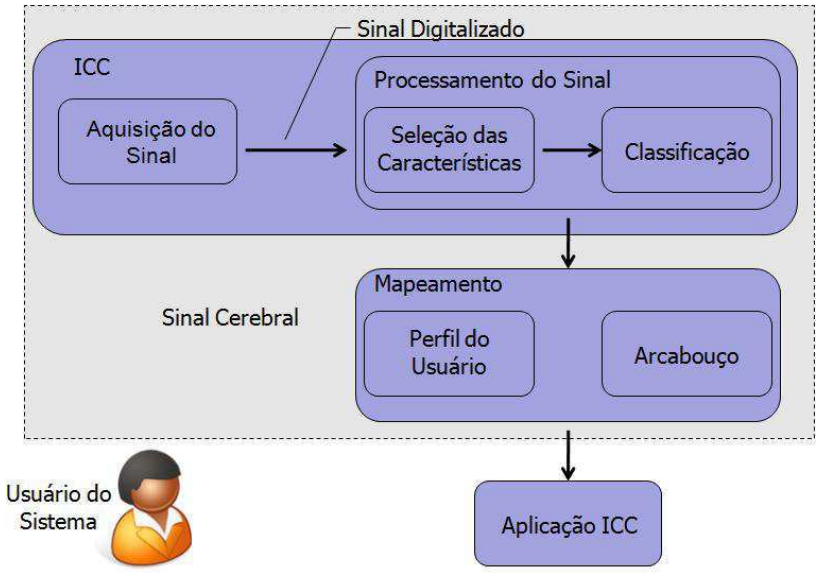

Figura 3.1: Diagrama de blocos da arquitetura introduzida neste trabalho

No módulo ICC, são realizadas as funcionalidades básicas de um sistema ICC. No Capítulo 2, foram discutidas essas funcionalidades. No módulo Mapeamento, os sinais cerebrais são mapeados em sinais de comando para dispositivos multimídia a partir das preferências do usuário, do dispositivo que se deseja controlar e com o auxílio do arcabouço de *software*. As

preferências do usuário são recuperadas pela aplicação a partir do perfil do usuário. Um exemplo de perfil do usuário é apresentado no Capítulo 4.

O arcabouço de *software* foi implementado um para facilitar o desenvolvimento de aplicações multimídia para sistemas ICC pervasivos. A estrutura do sistema desenvolvido, incluindo a arquitetura do arcabouço de *software*, é apresentada na figura 3.2.

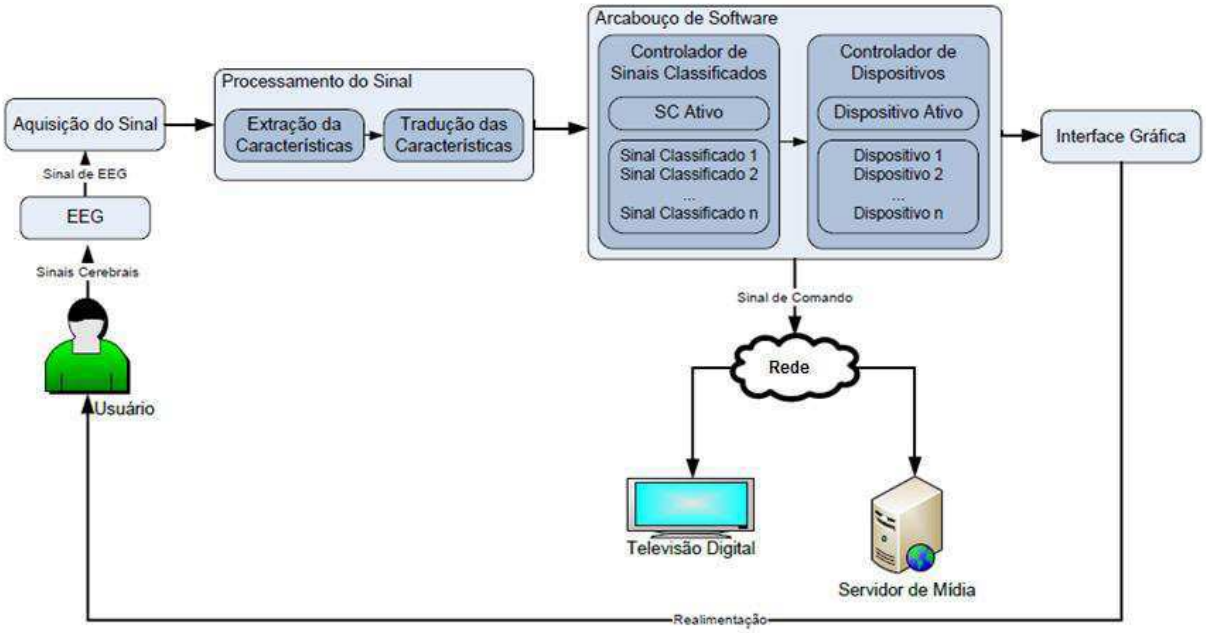

Figura 3.2: Sistema ICC pervasivo para controle de dispositivos multimídia

3.1 Cenário de Funcionamento do Sistema

Um dispositivo móvel foi colocado em frente ao usuário. Pediu-se ao participante que realizasse algumas tarefas, previamente definidas, para induzir alguns estados cerebrais desejados. Essas tarefas são discutidas na Seção 3.4. O cenário em que uma interface gráfica é apresentada ao participante enquanto ele realiza as tarefas pedidas, é ilustrado na Figura 3.3.

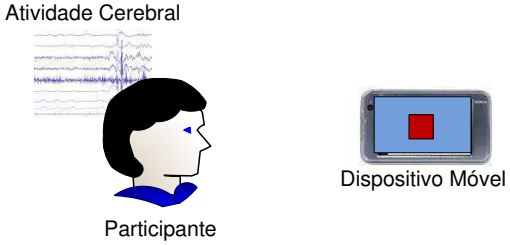

Figura 3.3: Representação do cenário em que é apresentada uma interface ao participante através de um dispositivo móvel para excitação de sua atividade cerebral

Os sinais cerebrais foram monitorados com o uso do método eletroencefalograma. Os eletrodos foram aplicados na superfície do escalpo do usuário para aquisição dos dados. Um número restrito de canais foi usado na leitura dos sinais para evitar o armazenamento de um grande número de dados que não seriam utilizados. A posição dos eletrodos e o monitoramento dos sinais cerebrais são discutidos na Seção 3.3. Na Figura 3.4 é apresentado o cenário onde os eletrodos são aplicados ao escalpo do participante.

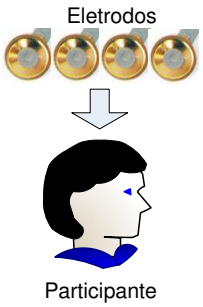

Figura 3.4: Representação do cenário em que os eletrodos são aplicados ao escalpo do usuário

Os dados adquiridos foram enviados através de cabos a um dispositivo embarcado no equipamento de EEG. Esse dispositivo é responsável pela amplificação, digitalização e extração das características dos sinais de EEG. As características selecionadas foram transmitidas para um dispositivo móvel através da tecnologia sem fio *Bluetooth*. O dispositivo móvel traduziu as características extraídas em sinais que representam as intenções do usuário. As etapas de processamento seleção e classificação das características do sinal são discutidas nas Seções 3.5.1 e 3.5.2, respectivamente. O cenário onde são ilustrados os dispositivos responsáveis pelo processamento do sinal é mostrado na Figura 3.5.

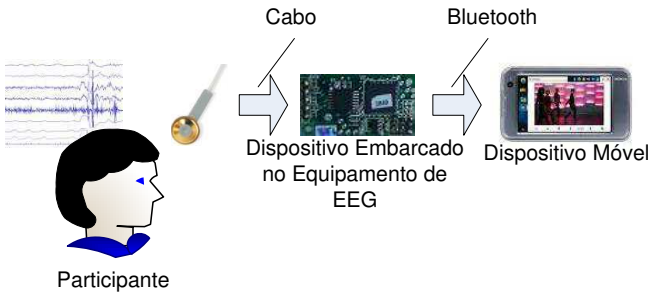

Figura 3.5: Representação do cenário que ilustra o processamento do sinal

No dispositivo móvel também foi implementado o arcabouço de *software*. O arcabouço de *software* foi usado para mapear as intenções do usuário em sinais de comando para aplicações multimídia. O desenvolvimento e funcionamento do arcabouço é discutido na Seção 3.7. O dispositivo móvel trata-se de um dispositivo de controle que implementa o protocolo UPnP-UP [46] para descoberta de dispositivos multimídia na rede e personalização de serviços. Os

dispositivos encontrados na rede pelo dispositivo de controle são listados para o usuário através de uma interface gráfica. O usuário foi automaticamente autenticado em um servidor de perfis após a seleção do dispositivo desejado. Uma aplicação multimídia personalizada foi apresentada ao usuário através de uma interface gráfica. A interação entre o dispositivo de controle e os dispositivos multimídia foi realizada através do protocolo UPnP-UP. Uma discussão sobre o acesso aos dispositivos multimídia é abordada na Seção 3.6. Na Figura 3.6 é ilustrado o cenário onde o dispositivo móvel procura dispositivos multimídia na rede e mapeia os sinais classificados em sinais de comando para o controle dos dispositivos.

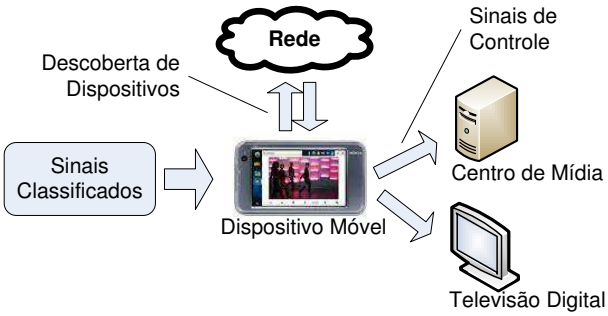

Figura 3.6: Representação do cenário que apresenta a descoberta de dispositivos multimídia na rede e o mapeamento do sinal classificado em um sinal de comando para o controle do dispositivo selecionado

3.2 Hardware

Um equipamento para aquisição dos sinais de EEG foi desenvolvido no Laboratório de Sistemas Embarcados e Computação Pervasiva da Universidade Federal de Campina Grande (UFCG) como parte deste trabalho. Apesar de existirem dispositivos de eletrônica de consumo de baixo custo disponíveis no mercado para o monitoramento dos sinais cerebrais, como o *MindSet* (199 dólares) fabricado pela *NeuroSky* e o *EPOC headset* (299 dólares) produzido pela *Emotiv*, optou-se pelo desenvolvimento do *hardware* para leitura dos sinais de EEG. Essa escolha foi devido ao fato de que os dispositivos citados não são adequados para serem aplicados no contexto deste trabalho. O *MindSet* possui apenas um canal para aquisição do sinal de EEG o que limita o desenvolvimento de aplicações que possam ser controladas por esse dispositivo. Apesar de ser maior o número de canais de aquisição do *EPOC headset*, os canais existentes são diferentes dos necessários para o desenvolvimento do sistema desenvolvido. Outro fator importante para a decisão de se desenvolver o equipamento de EEG foi o fato de que empresas como a *NeuroSky* vem produzindo chips que além de fazer a aquisição do sinal cerebral, implementam também funcionalidades para implementar grande parte do processamento do sinal. As operações de

processamento desempenhadas por esses chips incluem a amplificação, digitalização e seleção das características do sinal.

O equipamento implementado possui quatro canais para aquisição de sinais de EEG. Na implementação do *hardware* foram usados quatro módulos *ThinkGear* fabricados pela *NeuroSky*, sendo que cada módulo realiza a aquisição e parte do processamento do sinal referente a um dos canais. Optou-se por usar um módulo para cada canal de aquisição e não apenas um módulo para os quatro canais, para evitar o uso da eletrônica analógica para o tratamento do sinal de EEG, o que poderia descaracterizar o sinal, já que este possui amplitude bastante reduzida (da ordem de microvolts), devido, por exemplo, a interferência de outros sinais. Na Figura 3.7 é apresentada uma foto do módulo *ThinkGear* usado.

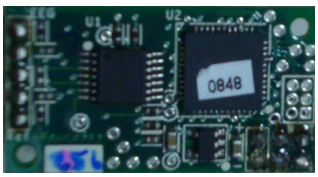

Figura 3.7: Foto do módulo *ThinkGear* usado

Um microcontrolador *Arduino Nano* [7] foi usado para multiplexar os sinais de EEG dos quatro módulos. O microcontrolador *Arduino Nano* foi escolhido devido as suas dimensões serem bastante compactas e a sua facilidade de uso. Na Figura 3.8 é apresentada uma foto do módulo usado.

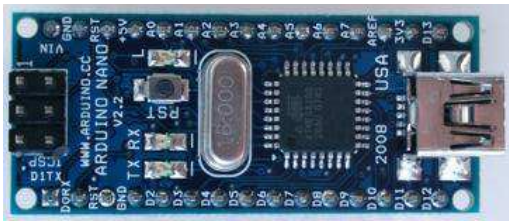

Figura 3.8: Foto do módulo usado

Um módulo de *Bluetooth* foi usado para realização da comunicação entre o Arduino Nano e o restante do sistema. A interface *Bluetooth* foi escolhida por ser uma tecnologia de comunicação sem fio de baixo custo e baixo consumo de bateria, o que possibilita uma maior mobilidade para o usuário do equipamento. Para a leitura dos sinais foram usados os eletrodos de superfície tipo concha, ou convexos, feitos de Ag/AgCl. Uma foto desses eletrodos é apresentada na Figura 3.9.

A implementação do *hardware* usado para aquisição dos sinais cerebrais é discutida com mais detalhes na Seção 4.1.

Figura 3.9: Foto do eletrodo de superfície tipo concha, ou convexo

3.3 Aquisição dos Sinais

Os sinais de EEG foram adquiridos pelo equipamento desenvolvido neste trabalho. Os eletrodos foram posicionados de acordo com o sistema Internacional 10-20. As localizações escolhidas, para aplicação dos eletrodos, foram O1 (região occipital do lado esquerdo do cérebro), FP1 (região frontopolar no hemisfério esquerdo cerebral), C3 (sulco central no lado esquerdo do cérebro) e C4 (sulco central no hemisfério direito do cérebro).

O eletrodo aplicado na posição O1 foi usado para identificar o fechamento e abertura dos olhos. O eletrodo colocado na posição FP1 foi usado no monitoramento do nível de atenção do usuário. Os eletrodos aplicados nas localizações C3 e C4 foram usados na identificação dos movimentos dos braços direito e esquerdo, respectivamente. Esses canais foram escolhidos para leitura do sinal pois é neles que as características que possuem maior relevância para este trabalho são mais marcantes durante o desempenho das tarefas citadas anteriormente. Os eletrodos de referência (Ref) e terra (Gnd) foram aplicados na orelha esquerda do usuário. Na Seção 4.2, a realização da aquisição dos sinais de EEG é abordada com mais detalhes. São ilustradas na Figura 3.10 as localizações dos eletrodos e suas respectivas funções.

3.4 Treinamento

O treinamento dos participantes foi dividido em duas etapas: uma para o aprendizado do estado cerebral durante os movimentos dos braços esquerdo e direito e outra para o aprendizado durante a abertura e fechamento dos olhos.

Na primeira etapa, inicialmente pediu-se ao participante que ficasse relaxado e que não executasse nenhum movimento. Após 3 segundos foi mostrada uma figura que representava o movimento do braço direito. Esta figura foi apresentada durante 2 segundos. O indivíduo foi instruído a executar ou imaginar o movimento do braço direito enquanto a figura estava sendo mostrada. Em seguida o sinal foi classificado e transformado em um estímulo para o usuário. O cenário usado para o treinamento do estado cerebral durante o desempenho do movimento do braço direito, é ilustrado na Figura 3.11.

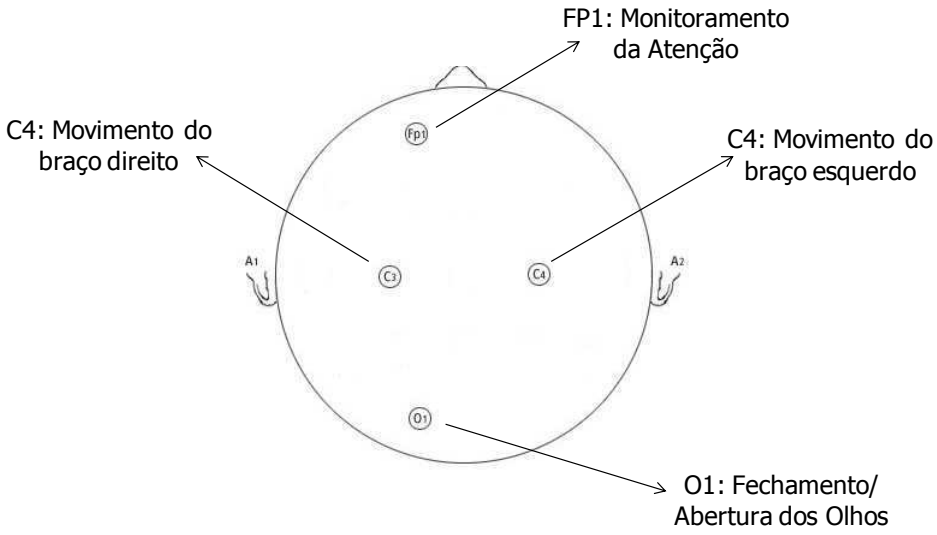

Figura 3.10: Localização dos eletrodos e suas respectivas funções

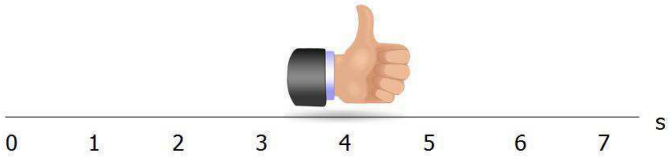

Figura 3.11: Cenário usado para o treinamento do estado cerebral durante o desempenho do movimento do braço direito

O mesmo procedimento foi realizado para a o treinamento do estado cerebral durante a execução do movimento do braço esquerdo. Cada um dos experimentos citados foi realizado vinte vezes.

A segunda etapa foi iniciada com os participantes realizando uma adaptação ao escuro durante alguns minutos. Após o processo de adaptação, os indivíduos foram instruídos a abrir os olhos durante 30 segundos e em seguida fechar durante 30 segundos. Esse cenário é apresentado na Figura 3.12. O ciclo citado foi repetido seis vezes. Após o término desse experimento, pediu-se aos usuários que eles fechassem os olhos por um período de 3 minutos.

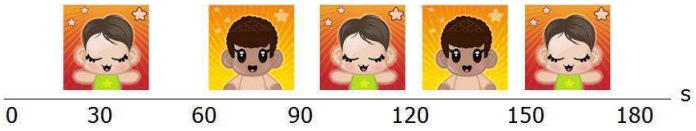

Figura 3.12: Cenário usado para o treinamento do estado cerebral durante o fechamento/abertura dos olhos

Na Seção 4.3 é discutida a realização da etapa de treinamento em detalhes.

3.5 Processamento

O processamento do sinal, como em todo sistema ICC, foi dividido em duas etapas: a seleção das características do sinal e a classificação dessas características.

3.5.1 Seleção das Características

A seleção das características do sinal de EEG foi realizada pelo módulo *ThinkGear*. Esse módulo foi usado na extração das componentes rítmicas cerebrais delta, teta, alfa, beta e gama. Neste trabalho, foram usadas apenas as componentes alfa por serem mais relevantes durante a execução das tarefas que deverão ser desempenhadas pelo usuário.

As ondas cerebrais alfa aumentam quando uma pessoa fecha os olhos, e diminuem quando ela abre os olhos. Durante períodos em que um indivíduo se encontra de olhos fechados, as ondas alfa aparecem fortemente na região occipital do cérebro. Uma redução nas características alfa é encontrada no córtex motor (sulco central) devido ao desempenho de um movimento ou a intenção de executá-lo. Um aumento no nível de atenção do usuário provoca uma redução das componentes alfa observadas na região frontopolar do cérebro.

3.5.2 Classificação

As componentes alfa dos sinais de EEG foram classificadas de acordo com o tipo de atividade realizada pelo usuário: abertura/fechamento dos olhos, conservação do estado de atenção e execução dos movimentos dos braços direito e esquerdo. Esta etapa foi realizada por um dispositivo inteligente com capacidade de processamento suficiente para executar um algoritmo de tradução.

O algoritmo de aprendizado máquina de suporte vetorial foi usado para tradução das características alfa em sinais que representam as intenções do usuário. Esse algoritmo foi escolhido por ser bastante usado em sistemas ICC. Para a escolha do algoritmo não foi realizado nenhum estudo comparativo entre diferentes métodos, pois a classificação das características do sinal não é o foco deste trabalho. A classificação das características do sinal e o método usado para classificá-las são discutidos na Seção 4.4.

3.6 Acesso aos Dispositivos Multimídia

O acesso aos dispositivos multimídia foi realizado através do arcabouço de *software* BRisa [3]. O BRisa permite que usuários descubram dispositivos multimídia, compartilhem, pesquisem e reproduzam conteúdos multimídia através de uma rede local ou da Internet. O Brisa foi escolhido como mecanismo de acesso aos dispositivos multimídia pelo fácil acesso aos desen-

volvedores desse arcabouço devido ao fato de que foi desenvolvido no Laboratório de Sistemas Embarcados e Computação Pervasiva da UFCG.

Foi usada a versão do BRisa que dá suporte ao protocolo UPnP-UP. O protocolo UPnP-UP foi criado com o objetivo de habilitar perfis de usuários para o padrão UPnP [12]. O uso de um protocolo que permita a autenticação de usuários é importante no contexto deste trabalho para disponibilização de serviços personalizados para o usuário. Os serviços são personalizados a partir do perfil do usuário que contém as suas preferências. A disponibilização de serviços personalizados em sistemas ICC foi usada para tornar esses sistemas mais rápidos no sentido de que o usuário tem que executar menos comandos para desempenhar uma determinada tarefa. Um exemplo disso pode ser observado no controle de um centro de mídia. Com o uso do protocolo UPnP-UP é possível que ao acessar um diretório de músicas, sejam apresentas ao usuário apenas as suas músicas preferidas. Isso ajuda na seleção da música pelo usuário que necessitou realizar menos operações de interação para executar a música desejada.

3.7 Arcabouço de Software

O arcabouço de *software* foi desenvolvido com o objetivo de auxiliar desenvolvedores na implementação de aplicações ICC pervasivas. Esse arcabouço possui uma arquitetura baseada em componentes de *software*. Esse tipo de arquitetura foi escolhido porque ele traz algumas vantagens para o desenvolvedor de componentes. O desenvolvedor de componentes não precisa se preocupar com as interfaces de comunicação com o sistema, mas sim apenas com seus algoritmos. O arcabouço é composto por dois controladores: o controlador de sinais classificados e o controlador de dispositivos. A arquitetura do arcabouço de *software* é apresentada na Figura 3.13.

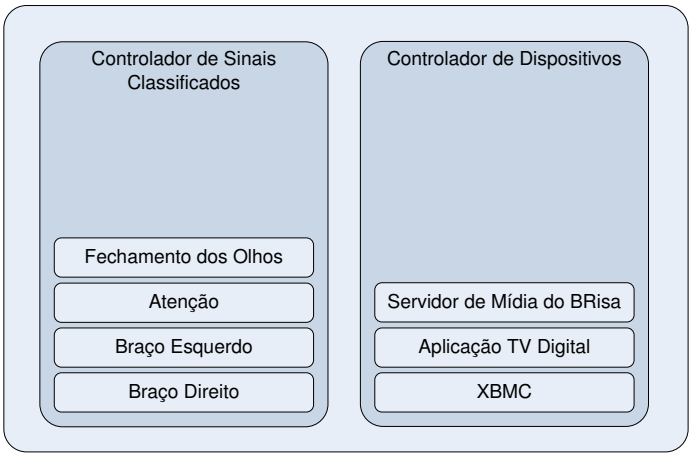

Figura 3.13: Arquitetura do Arcabouço de *Software*

O controlador de sinais classificados mapeia o sinal classificado em um evento. Esse ma-

peamento depende do sinal classificado e do dispositivo multimídia que será controlado. Neste controlador cada componente de *software* é referente a um tipo diferente de sinal classificado. Os componentes de *software* já existentes no controlador de sinais classificados são: o movimento do braço direito, o movimento do braço esquerdo, o fechamento/abertura dos olhos e o nível de atenção. Caso o desenvolvedor de uma aplicação necessite usar um sinal classificado diferente dos citados, ele deve desenvolver um componente referente a este novo sinal.

O controlador de dispositivos também é composto por componentes. Cada componente é referente a um tipo de dispositivo multimídia. Alguns componentes já foram previamente desenvolvidos. Um exemplo desses componentes já existentes é o componente referente ao centro de mídias XBMC [6]. Caso o desenvolvedor deseje desenvolver uma aplicação para um dispositivo diferente dos que já possuem um componente, ele deve implementar um novo componente de *software*. A implementação do arcabouço é discutida na Seção 4.6.

Quando um desenvolvedor deseja implementar uma aplicação usando o arcabouço de *software* desenvolvido neste trabalho, ele deve seguir algumas regras que dependem de qual cenário a aplicação que ele deseja desenvolver está inserida. Esses cenários são divididos em três categorias: não existem implementados no arcabouço um ou mais componentes relativos aos sinais classificados que serão usados pela aplicação, não existe o componente relativo ao dispositivo que será controlado ou uma composição do primeiro e segundo cenário.

No primeiro caso, quando não existem um ou mais componentes relativos aos sinais classificados que serão usados pela aplicação, mas existe o componente relativo ao dispositivo que se deseja interagir, é necessário que o desenvolvedor se preocupe apenas com a implementação de um componente para cada sinal classificado inexistente. Na Seção 3.7.1 é discutido o procedimento a ser seguido para o desenvolvimento de um componente relativo a um sinal classificado.

No segundo caso, quando já existem implementados no arcabouço os componentes relativos aos sinais classificados, mas não existe o componente relativo ao dispositivo multimídia a ser usado pela aplicação, o desenvolvedor deve implementar o componente referente ao dispositivo desejado. Na Seção 3.7.1 é abordado o processo de implementação de um componente referente a um dispositivo multimídia.

Por último, se não existem implementados no arcabouço um ou mais componentes referentes aos sinais classificados e nem o componente relativo ao dispositivo multimídia desejado, o desenvolvedor terá que implementar os componentes relativos aos sinais classificados inexistentes como explicado no primeiro caso. Além disso, o desenvolvedor também deverá implementar o componente relativo ao dispositivo multimídia desejado, como explicado no segundo caso.

É importante destacar que o arcabouço não trará nenhuma vantagem ao desenvolvedor que deseje implementar uma aplicação onde nenhum dos sinais classificados a serem usados possuam componentes relativos a eles e nem exista um componente relativo ao dispositivo multi-

mídia desejado.

3.7.1 Desenvolvimento de Componentes

Como o arcabouço de *software* foi desenvolvido usando o arcabouço *PyQt* [10], é necessário que o desenvolvedor de componentes configure o ambiente de desenvolvimento *PyQt*, antes de iniciar a implementação de um componente. Para isso, o desenvolvedor deve instalar algumas ferramentas. As ferramentas que devem ser instaladas dependem da plataforma usada na implementação do componente. Em [49] é discutido quais ferramentas devem ser instaladas e como deve ser realizada a instalação dessas ferramentas em diversas plataformas, como Windows, Mac OS X, Unix e Linux, para que o ambiente de desenvolvimento *PyQt* seja configurado.

Depois de configurar o ambiente de desenvolvimento *PyQt*, o desenvolvedor deve fazer o *download* do arcabouço de *software* desenvolvido neste trabalho. Em seguida o desenvolvedor pode iniciar a implementação de um componente. Existem dois tipos de componentes: os componentes relativos aos sinais de EEG classificados e os componentes relativos aos dispositivos multimídia a serem controlados. Nas próximas seções é discutido o desenvolvimento dos dois tipos de componentes.

Componentes Relativos aos Sinais de EEG Classificados

Para implementação de um componente relativo a um sinal de EEG classificado, o desenvolvedor deve seguir os seguintes passos:

- Criar um módulo *python* referente ao sinal de EEG classificado.
- Implementar uma classe que receba como parâmetro o dispositivo multimídia que está sendo controlado.
- Implementar um método que tenha a função de mapear o sinal classificado em um evento para o dispositivo multimídia.
- Implementar um método que transmita o evento para o módulo *python*, denominado *controladorDeSinais*.
- Importar o componente desenvolvido no módulo *controladorDeSinais*.
- Implementar a condição em que o sinal classificado corresponde ao novo componente na função *carregaComponente()* da classe *controladorSinaisClassificados*.
- Na condição implementada no método *carregaComponente()*, instanciar o componente.

Exemplos que ilustram o passo a passo de como devem ser desenvolvidos os componentes relativos aos sinais de EEG classificados são apresentados mais adiante na Seção 4.6.4.

Componentes Relativos aos Dispositivos Multimídia

Para implementação de um componente relativo à um dispositivo multimídia, o desenvolvedor deve seguir os seguintes passos:

- Criar um módulo *python* referente ao dispositivo multimídia a ser controlado.
- Implementar uma classe relativa ao componente.
- Implementar um método que receba como parâmetro um evento para o dispositivo multimídia, e que tenha a função de mapear esse evento em um comando para o dispositivo.
- Implementar um método para cada comando a ser usado no controle do dispositivo multimídia. Cada um desses métodos tem a função de enviar o comando ao dispositivo.
- Importar o componente desenvolvido no módulo *controladorDeDispositivos*.
- Implementar a condição em que o dispositivo selecionado corresponde ao novo componente na função *carregaComponente()* da classe *controladorDispositivos*.
- Na condição implementada no método *carregaComponente()*, instanciar o componente.

Exemplos que ilustram passo a passo de como devem ser desenvolvidos os componentes relativos aos dispositivos multimídia são apresentados mais adiante na Seção 4.6.5.

3.7.2 Desenvolvimento de Aplicações

O desenvolvimento de uma aplicação requer a configuração do ambiente de desenvolvimento *PyQt*. A configuração deve ser realizada da mesma forma que é feita para o desenvolvimento de um componente como foi discutido na Seção 3.7.1.

Depois de configurar o ambiente de desenvolvimento *PyQt*, o desenvolvedor deve realizar os seguintes passos:

- Realizar o *download* do arcabouço de *software* desenvolvido neste trabalho.
- Realizar o *download* do BRisa e suas dependências. Os pacotes relativos ao BRisa e suas dependências são disponibilizados em [3].
- Instalar o BRisa e suas dependências.
- Se certificar de que todos os componentes necessários para o desenvolvimento da aplicação já estão implementados pelo arcabouço. No caso de estar faltando algum componente, o desenvolvedor deve implementá-lo seguindo os passos discutidos na Seção 3.7.1.

- Criar um módulo *python* para a implementação de um dispositivo de controle. Esse dispositivo deve realizar a descoberta de dispositivos multimídia na rede usando o BRisa. A interface gráfica principal do dispositivo de controle deve apresentar uma lista com os dispositivos disponibilizados na rede e uma legenda que ajude o usuário a identificar qual tarefa deve desempenhar para executar um determinado comando. O ponto de controle deve ser desenvolvido em *PyQt*. Uma discussão sobre o dispositivo de controle é abordada na Seção 4.5

- Desenvolver uma segunda interface gráfica para o dispositivo de controle em *PyQt*. Essa segunda interface deve ser carregada após a seleção do dispositivo desejado, pelo usuário do sistema. Essa interface deve ser desenvolvida baseada nos comandos a serem usados no controle do dispositivo selecionado e é usada para realimentação do usuário.

- Importar nos módulos criados para o desenvolvimento de ambas as interfaces, as bibliotecas *QtCore* e *QtGui* de *PyQt*.

- Da mesma forma que no desenvolvimento da interface gráfica principal, implementar na segunda interface gráfica uma legenda com as tarefas que o usuário deve desempenhar para que ele consiga executar os comandos para o controle do dispositivo multimídia . Além disso, desenvolver mecanismos que mostrem ao usuário que ele manteve determinado estado cerebral no desempenho de uma dada tarefa e que obteve sucesso em sua execução.

- Importar o módulo *principal* do arcabouço nos módulos das interfaces gráficas.

- Criar uma instância da classe *Principal* nas classes que implementam as interfaces gráficas.

- Usar o mecanismo sinais e *slots* de *PyQt* para conectar as interfaces gráficas com o arcabouço. Esse mecanismo é usado para que a interface fique aguardando a chegada do sinal de atualização da interface.

- Implementar um método que atualize a interface de acordo com o sinal de atualização recebido.

Um exemplo que ilustra o passo a passo a ser seguido para o desenvolvimento de uma aplicação utilizando o arcabouço é apresentado na Seção 4.6.6.

3.7.3 Funcionamento do Arcabouço de Software

Uma interface gráfica com uma lista de dispositivos é apresentada ao usuário do sistema. O usuário deve selecionar nessa lista o dispositivo que ele deseja controlar. Após a seleção do

dispositivo, uma mensagem com o dispositivo selecionado é enviada ao arcabouço. A representação do cenário em que o arcabouço recebe uma mensagem indicando o dispositivo selecionado, nesse caso o centro de mídias XBMC, é mostrada na Figura 3.14.

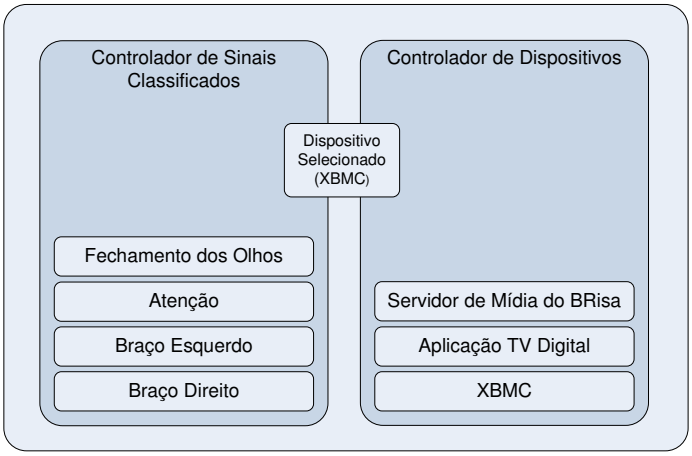

Figura 3.14: Representação do cenário em que o arcabouço recebe a mensagem com o dispositivo selecionado

Depois de receber a mensagem com o dispositivo selecionado, o controlador de dispositivos seleciona o componente referente ao dispositivo multimídia selecionado e fica esperando a chegada de eventos para este dispositivo. Na Figura 3.15 é mostrado o arcabouço após o carregamento do componente referente ao dispositivo selecionado.

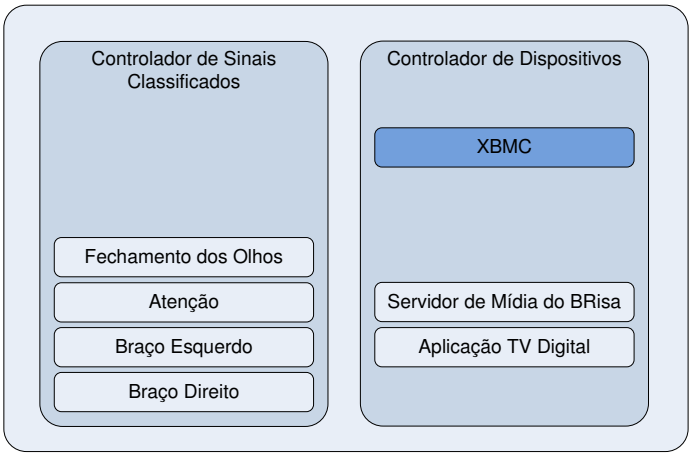

Figura 3.15: Representação do cenário em que o componente do centro de mídias XBMC é carregado pelo controlador de dispositivos

Quando o usuário realiza uma das tarefas já discutidas neste trabalho, seu estado cerebral é classificado em um tipo de sinal. Por exemplo, quando o usuário imagina ou realiza o movimento do braço direito, sua atividade cerebral é classificada em um sinal que representa o movimento do braço direito. Ao desempenhar uma dessas tarefas um sinal classificado é enviado ao

arcabouço. Uma representação do cenário em que o sinal que representa o movimento do braço direito é transmitido ao arcabouço é mostrada na Figura 3.16.

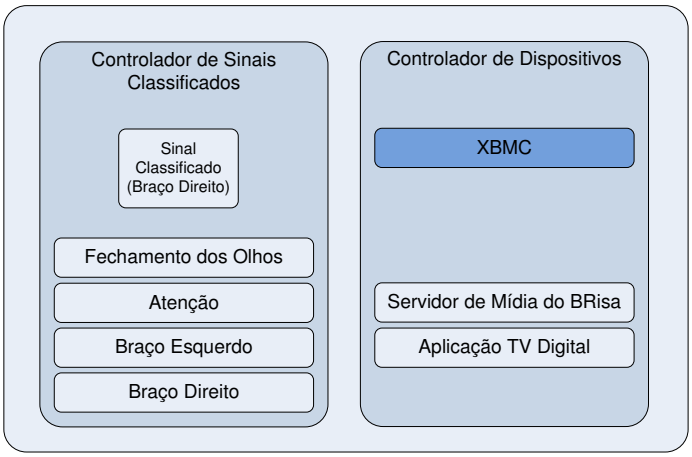

Figura 3.16: Representação do cenário em que o sinal classificado que representa o movimento do braço direito é enviado ao arcabouço

Após um sinal classificado ser recebido pelo arcabouço, o controlador de sinais classificados seleciona o componente referente a esse sinal. Na Figura 3.17 é apresentado o cenário em que o controlador de sinais classificados carrega o componente referente ao sinal classificado como movimento do braço direito após esse sinal ser recebido.

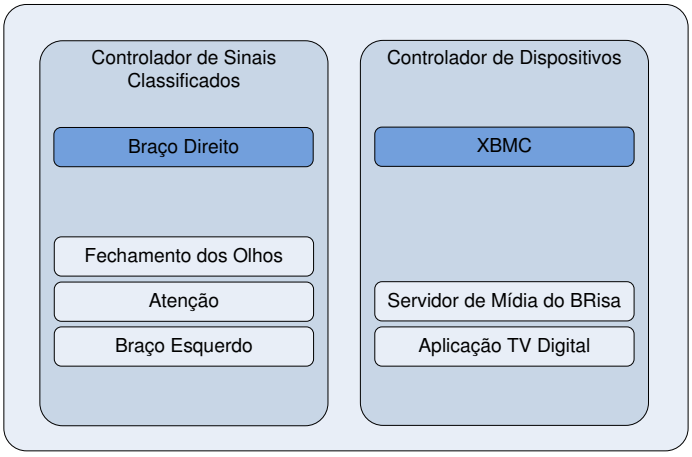

Figura 3.17: Representação do cenário em que o componente do sinal que representa o movimento do braço direto é carregado pelo controlador de sinais classificados

O componente carregado pelo controlador de sinais classificados mapeia o sinal classificado em um evento para o dispositivo multimídia selecionado e transmite esse evento ao componente relativo ao dispositivo selecionado. Esse cenário é apresentado na Figura 3.18.

Após transmitir o evento ao componente relativo ao dispositivo multimídia selecionado, o componente referente ao sinal classificado é desativado pelo controlador de sinais classificados. Na Figura 3.19 é mostrado o cenário em que o componente relativo ao sinal classificado

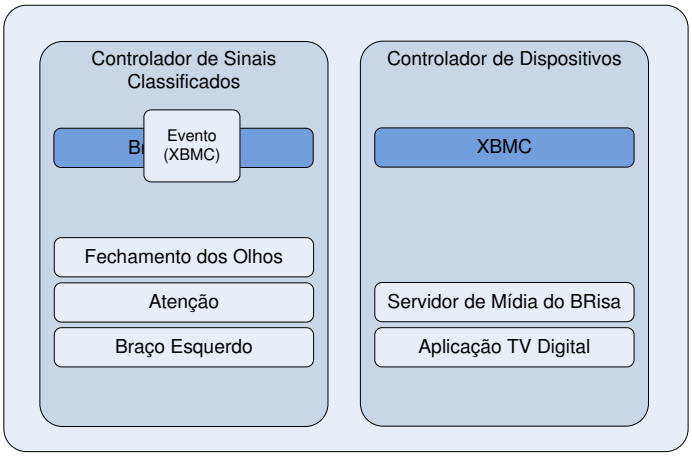

Figura 3.18: Representação do cenário em que o componente do sinal que representa o movimento do braço direto mapeia o sinal classificado em um evento para o dispositivo selecionado

como movimento do braço direito é descarregado após transmitir o evento para o componente referente ao centro de mídias XBMC.

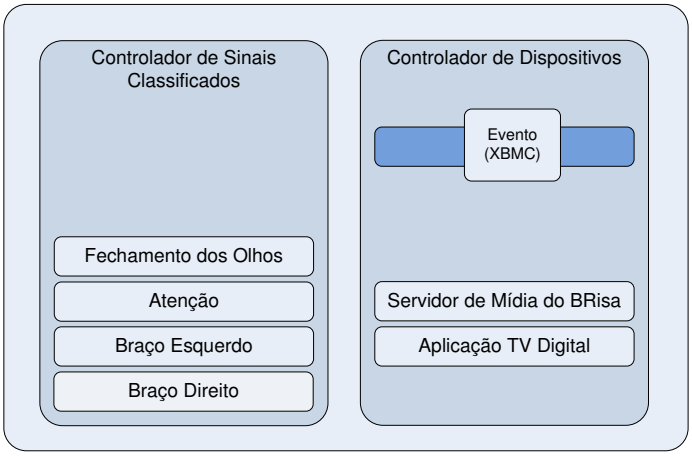

Figura 3.19: Representação do cenário em que o componente do sinal que representa o movimento do braço direto é descarregado após transmitir um evento para o dispositivo selecionado

O componente relativo ao dispositivo selecionado recebe o evento e o mapeia em um sinal de comando para o dispositivo multimídia e em um sinal de atualização para a interface gráfica usada para realimentar o usuário. Na Figura 3.20 é apresentado o cenário em que o componente relativo ao centro de mídias XBMC mapeia o evento recebido em um sinal de comando e na Figura 3.21 é mostrado o cenário em que o mesmo componente mapeia o evento recebido em um sinal de atualização para a interface gráfica.

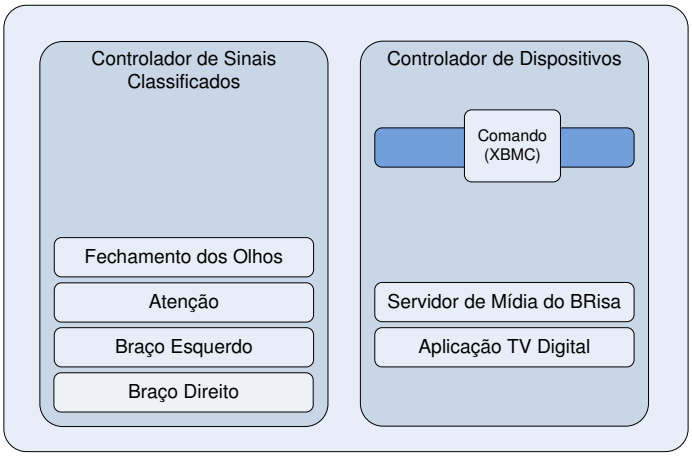

Figura 3.20: Representação do cenário em que o componente relativo ao dispositivo selecionado mapeia o evento recebido em um comando para o centro de mídias XBMC

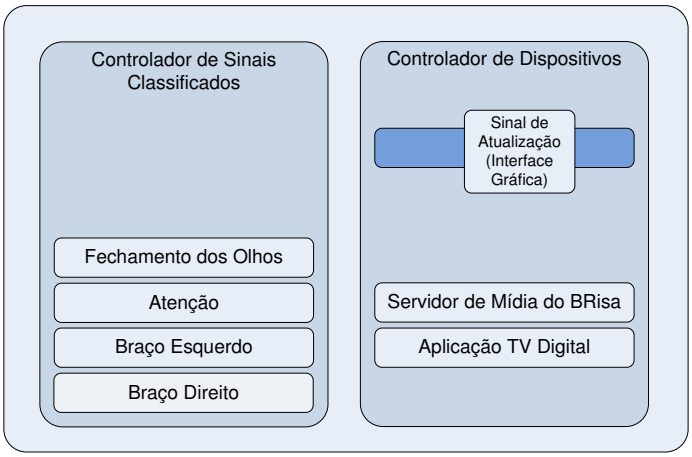

Figura 3.21: Representação do cenário em que o componente relativo ao dispositivo selecionado mapeia o evento recebido em um sinal de atualização para a interface gráfica usada para realimentar o usuário

3.8 Realimentação

A realimentação do usuário do sistema é realizada visualmente através da interface gráfica das aplicações multimídia. Na interface são mostrados os estados cerebrais que o usuário deve manter para executar um determinado comando. Além disso, quando o usuário realiza alguma atividade a interface é atualizada para realimentar o participante. Na Seção 4.6.6 é apresentada uma ilustração de uma interface gráfica usada na realimentação do usuário.

3.9 Teste de Usabilidade

O objetivo nos testes de usabilidade a serem realizados no contexto deste trabalho, é avaliar o sistema ICC pervasivo para o controle de dispositivos multimídia. O teste de usabilidade é

um processo no qual alguns avaliadores analisam o uso de um determinado produto pelos seus usuários. Durante o teste, o usuário desempenha uma variedade de tarefas, enquanto inspetores coletam informações relativas ao processo de interação do usuário. Essas informações incluem os erros cometidos pelo usuário, o conforto do usuário durante a realização de uma tarefa, quando e onde o usuário confunde-se ou frustra-se, a rapidez e a satisfação com que o usuário realiza uma tarefa [21].

Os testes de usabilidade são realizados com o objetivo de avaliar a usabilidade de um produto. Segundo a norma ISO 9241, usabilidade é a medida pela qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com efetividade, eficiência e satisfação em um contexto de uso específico [31]. As avaliações de usabilidade a serem realizadas foram planejadas e elaboradas de acordo com o protocolo experimental desenvolvido no LIHM (Laboratório de Interface Homem-Máquina da Universidade Federal de Campina Grande). A aplicação de um protocolo em ensaios de usabilidades tem o objetivo de facilitá-los.

Escolheu-se o protocolo experimental desenvolvido no LIHM [14] para ser aplicado aos testes de usabilidade devido, principalmente, ao fácil acesso aos seus desenvolvedores, que podem auxiliar na aplicação do protocolo aos ensaios.

O protocolo consiste de um conjunto de etapas, processos, e atividades que devem guiar os avaliadores no planejamento, condução, análise e elaboração do relatório do experimento.

O protocolo é dividido em módulos para facilitar seu uso e evitar que tempo e esforço sejam gastos desnecessariamente, pois em seu processo de aplicação existem muitos processos e atividades a serem desenvolvidos. Dependendo do produto e do seu contexto de uso alguns desses processos e atividades são desnecessários para avaliação do produto.

O protocolo consiste do "protocolo base" e dos "módulos de extensão". O protocolo base contém os processos e atividades essenciais ao protocolo. Os módulos de extensão incluem as atividades adicionais às presentes no protocolo base. Os módulos são definidos de acordo com: a fase de desenvolvimento do produto, a natureza do produto e a natureza do teste.

Com relação a fase de desenvolvimento do produto, o sistema desenvolvido no contexto deste trabalho é considerado como um protótipo do sistema final. Quanto à natureza do produto, o sistema ICC pervasivo é categorizado como para ser usado por indivíduos portadores de deficiência motora severa. Finalmente, com relação à natureza dos testes, os ensaios são divididos em duas categorias: em laboratório e em campo. Em uma primeira etapa, as avaliações devem ser realizadas em laboratório, tendo como usuários pessoas saudáveis. Em uma segunda etapa, devem ser realizados testes em campo, tendo como usuários pessoas portadoras de deficiência física severa, mas com o sistema cognitivo intacto.

Os testes a serem realizados em laboratório devem ser feitos no LIHM-UFCG em Campina Grande, enquanto que os testes a serem realizados em campo devem ser realizados no Instituto de Neurociências da Paraíba em João Pessoa, com os pacientes e o auxílio do médico neuro-

cirurgião, Maurus Holanda.

3.10 Sumário

No desenvolvimento de uma arquitetura e um arcabouço de software para auxiliar na implementação de sistemas ICC pervasivos tem-se o objetivo de motivar desenvolvedores a projetar esse tipo de sistema. Com a ampliação do número de aplicações ICC pervasivas e a disponibilização de equipamentos de aquisição e dispositivos móveis de baixo custo e serviços pervasivos voltados para aplicações ICC, é possível o uso de sistemas ICC pervasivos por pessoas portadoras de deficiência motora severa em qualquer lugar e a qualquer hora. O uso desses sistemas no dia-a-dia das pessoas citadas anteriormente fornece mais independência a esses usuários, e conseqüentemente uma melhora na qualidade de vida.

O método de aquisição, o equipamento usado para leitura dos sinais, o dispositivo usado para implementação do algoritmo de seleção e do arcabouço foram escolhidos para fornecer a maior mobilidade possível ao usuário do sistema. O uso de um protocolo que disponibiliza serviços personalizados para a comunicação entre o sistema e os dispositivos multimídia foi escolhido para diminuir o número de interações entre o usuário e s sistema durante a realização de uma determinada tarefa. Com isso, tem-se um sistema mais rápido e fácil de usar. Além disso, optou-se por planejar e elaborar um teste de usabilidade para avaliar o sistema, apesar de não terem sido realizados os testes no contexto deste trabalho devido a restrição de tempo, para detectar os principais problemas encontrados pelo usuário durante seu uso e com isso tentar melhorar o sistema.

Capítulo 4

Resultados

Neste capítulo é discutido como foi realizada a implementação de cada módulo do sistema ICC pervasivo desenvolvido neste trabalho. Essa discussão é iniciada com uma apresentação sobre o desenvolvimento do *hardware* usado na aquisição dos sinais, sendo mostrados os dispositivos usados, a interface de comunicação usada entre esses dispositivos e os requisitos para o equipamento implementado funcionar adequadamente. A discussão segue com a apresentação dos passos realizados para aquisição dos sinais cerebrais. Depois, é abordado o treinamento realizado com os participantes para que eles aprendam a operar o sistema, sendo mostrado o procedimento adotado durante as etapas do treinamento. O processamento do sinal é mostrado com foco na implementação do algoritmo máquina de suporte vetorial usado na classificação das características do sinal. Em seguida, é apresentada a realização do acesso aos dispositivos multimídia. É discutido o desenvolvimento do arcabouço de software, sendo apresentados os módulos desse arcabouço e os passos realizados para suas implementações. Finalmente, são abordados o planejamento e a elaboração do teste de usabilidade a ser realizado para avaliação do sistema ICC pervasivo para o controle de dispositivos multimídia.

4.1 Hardware

Como já discutido no capítulo 3, o equipamento para aquisição dos sinais de EEG desenvolvido no contexto deste trabalho é composto por quatro módulos *ThinkGear*, um microcontrolador *Arduino Nano* e um módulo *Bluetooth*. O diagrama de blocos do circuito implementado, é apresentado na Figura 4.1 e o diagrama esquemático é mostrado na Figura 4.2.

A comunicação entre os módulos *ThinkGear* e o microcontrolador é realizada através de uma interface serial. Os módulos *ThinkGear* são programáveis, e para que eles desempenhem o comportamento desejado são necessários que comandos de configuração sejam enviados para eles funcionarem adequadamente. O formato do comando de configuração é mostrado na Figura 4.3.

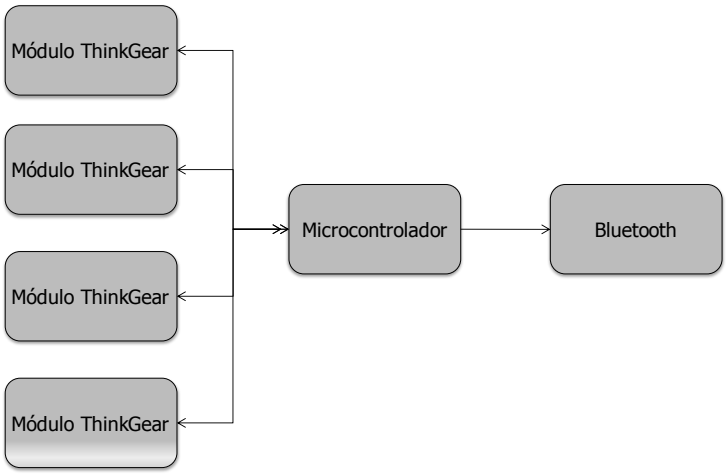

Figura 4.1: Diagrama de blocos do circuito de aquisição dos sinais de EEG

Para o circuito operar de acordo com o desejado, foi necessário habilitar a transmissão dos pacotes referentes aos valores das componentes delta, teta, alfa, beta e gama e aos níveis de atenção. Para habilitar a transmissão dos pacotes citados, foi necessário programar o módulo *ThinkGear* através de dois comandos: o 0x2A e o 0x3C. O primeiro comando apresentado é usado para configurar as saídas a serem lidas. O último comando é usado para configurar as saídas do *eSense*. O *eSense* é uma tecnologia desenvolvida pela *NeuroSky* que usa algoritmos para transformar os sinais cerebrais em níveis de atenção e meditação do usuário. Sempre que o circuito é iniciado esses comandos são transmitidos aos módulos *ThinkGear* para que estes sejam configurados.

O microcontrolador *Arduino Nano* possui apenas uma interface serial em *hardware*. Por esse motivo, para o *Arduino Nano* se comunicar com os quatro módulos *ThinkGear* através da interface serial foi necessária a implementação de mais quatro portas seriais em *software*. Para o desenvolvimento dessas novas interfaces seriais foi usada a biblioteca *NewSoftSerial* [2]. A listagem com o código usado para implementação das portas seriais em *software* é apresentada na Figura 4.4.

O *Arduino Nano* lê da porta serial 1, o pacote disponibilizado pelo módulo *ThinkGear* usado na aquisição do sinal a a partir do eletrodo localizado em O1. Depois da leitura o microcontrolador transmite esse pacote para o módulo *Bluetooth* através da porta serial implementada em *hardware*. Em seguida o pacote referente ao sinal adquirido pelo eletrodo localizado em C3 é lido da porta serial 2 e transmitido para o módulo *Bluetooth* da mesma forma que o pacote citado anteriormente. O próximo pacote é lido da porta Serial 3 e foi adquirido pelo eletrodo localizado em C4. Por último, é realizada a leitura do pacote monitorado pelo eletrodo localizado em FP1 a partir da serial 4. O terceiro e o quarto pacote são transmitidos para o módulo *Bluetooth* da mesma forma que o primeiro e o segundo pacotes.

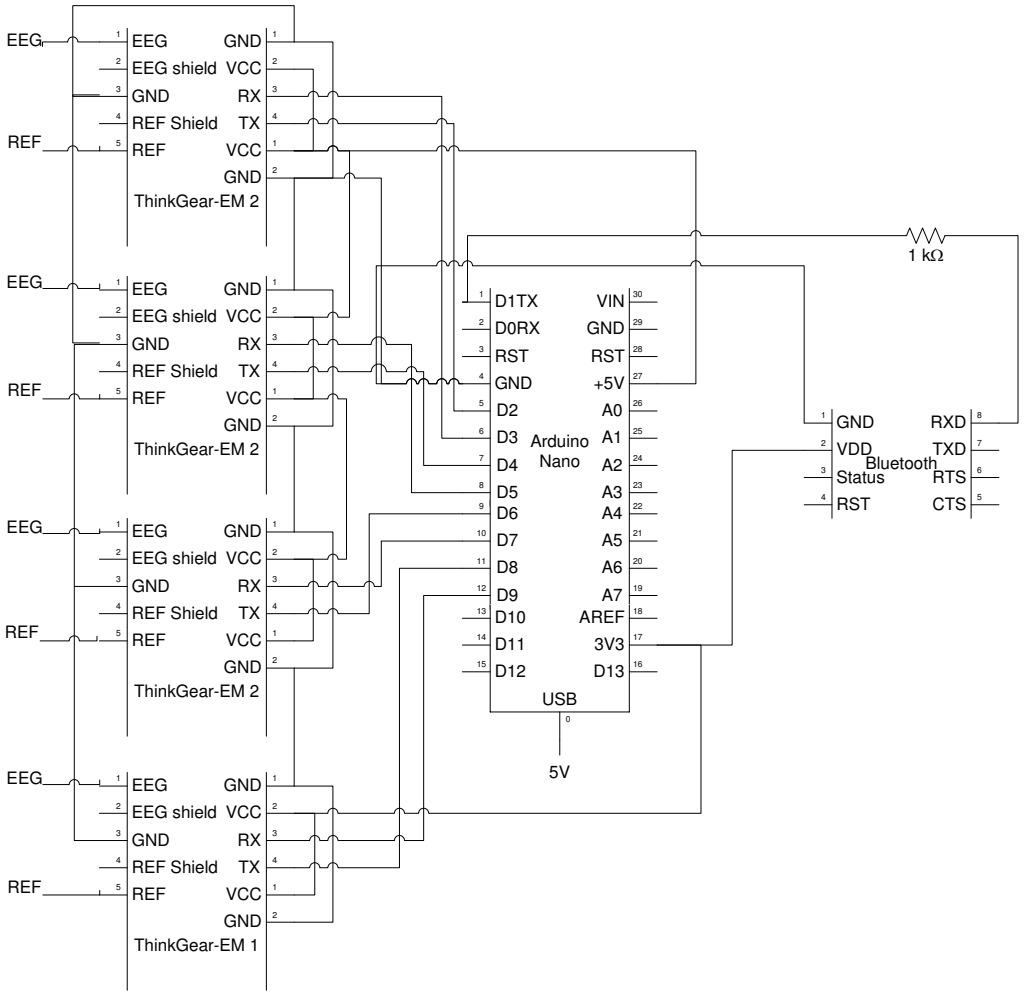

Figura 4.2: Diagrama esquemático do Circuito de Aquisição dos Sinais de EEG

4.2 Aquisição do Sinal Cerebral

Para aquisição dos sinais de EEG foi usado o equipamento desenvolvido neste trabalho. Antes de aplicar os eletrodos à superfície do escalpo do usuário do sistema, foi usado um multímetro para testar se os eletrodos estavam funcionando adequadamente. Além disso, foi verificado se os fios dos eletrodos estavam devidamente blindados para evitar ruído no sinal. O próximo passo foi aplicar os eletrodos ao couro cabeludo do usuário nas posições discutidas na Seção 3.3. Para a aplicação dos eletrodos foi necessário o uso de uma pasta condutora de EEG. Esta pasta foi usada para ajudar na condutividade dos sinais bioelétricos cerebrais e na fixação dos eletrodos na superfície do escalpo do participante. Os eletrodos foram fixados na cabeça do usuário através de um chapéu.

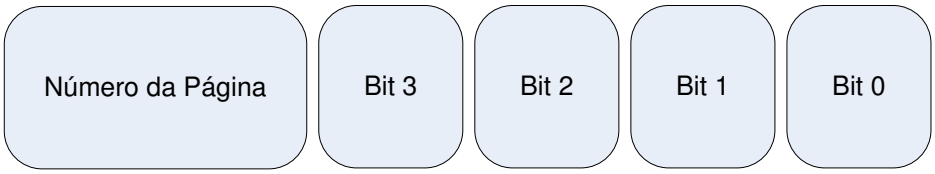

Figura 4.3: Formato do comando de configuração

```
#include <NewSoftSerial.h>

NewSoftSerial Serial1(2, 3);
NewSoftSerial Serial2(4, 5);
NewSoftSerial Serial3(6, 7);
NewSoftSerial Serial4(8, 9);
```

Figura 4.4: Listagem de código para implementação de 4 portas seriais através de *software*

4.3 Treinamento

Foi realizada uma etapa de treinamento para observar se os sinais de EEG se comportariam como o esperado durante o desempenho das tarefas de execução dos braços direito e esquerdo e fechamento/abertura dos olhos. O treinamento foi dividido em duas etapas: uma para o usuário aprender a controlar seu estado cerebral durante a execução dos braços direito e esquerdo e a outra para o fechamento/abertura dos olhos. Foi desenvolvido um *software* para cada uma das etapas. Mais adiante é discutida cada uma das etapas. Para realização do treinamento foi usada uma sala sem nenhum equipamento eletrônico, além do dispositivo inteligente usado para armazenar os sinais adquiridos, com o objetivo de evitar interferência de outros sinais no sinal cerebral. O período de realização do experimento teve a duração média de 1 hora e meia. Este tempo incluiu a aplicação dos eletrodos, as paradas entre o desempenho das tarefas pelo usuário e o tempo de preparação para o experimento.

Antes de iniciar o treinamento, pediu-se que o participante sentasse em uma cadeira confortável e que ficasse bastante relaxado evitando realizar movimentos que atrapalhassem a análise do sinal. Além disso, o usuário foi instruído a manter o foco no centro da tela do Internet *Tablet* Nokia N810 durante todo o experimento. Após três segundos foi apresentada ao usuário uma figura de um braço direito no centro da tela do N810. A tela onde é ilustrado esse cenário é apresentada na Figura 4.5.

Esta apresentação durou um período de dois segundos. Pediu-se ao usuário que enquanto a figura estivesse sendo mostrada na tela, ele desempenhasse a tarefa que estava sendo indicada pela figura, neste caso que o participante executasse o movimento do braço direito. Os últimos três segundos foram usados para o processamento do sinal. Durante o processamento, o sinal

Figura 4.5: Tela usada para o treinamento do estado cerebral durante a execução do movimento do braço direito

foi classificado em um movimento do braço direito e transformado em um estímulo visual que foi apresentado ao participante em forma de uma barra que teve um crescimento para o lado direito. Esse estímulo visual indica que a tarefa foi desempenhada com sucesso. Na Figura 4.6 é mostrada a tela onde é apresentado o estímulo visual.

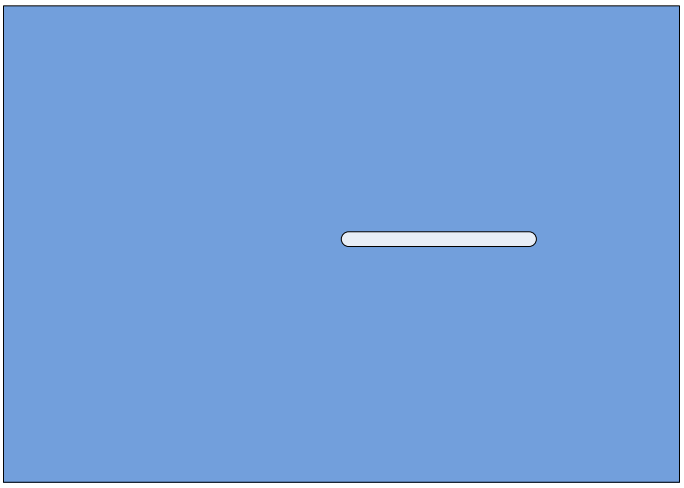

Figura 4.6: Tela usada para apresentar o estímulo visual ao participante após a execução do movimento do braço direito

Depois de um período entre um e três segundos foi apresentada uma nova figura ao participante, podendo ser novamente uma figura com o braço direito ou uma figura com o braço esquerdo. O período entre a coleta de uma nova amostra do sinal e a escolha da figura que é apresentada são aleatórios para evitar erros devido a uma adaptação durante a classificação do sinal. No caso de uma figura com o braço esquerdo ser apresentada ao usuário, o sinal classificado é transformado em um estímulo visual em forma de barra com crescimento para o lado esquerdo. Foram coletadas quarenta amostras do sinal, vinte amostras referentes a execução do

movimento do braço direito e vinte referentes ao movimento do braço esquerdo.

Após o término da primeira fase do treinamento, o participante foi instruído a ficar relaxado novamente, evitando realizar qualquer movimento. Além disso, pediu-se que o usuário fechasse os olhos por um período de um minuto para adaptação ao escuro. O fim desse período de adaptação ao escuro foi sinalizado ao usuário através de um *beep*. O usuário foi instruído a manter os olhos abertos, após o *beep*, focando em uma figura apresentada na tela do N810 durante um período de trinta segundos. A tela usada nesta etapa do treinamento é ilustrada na Figura 4.7.

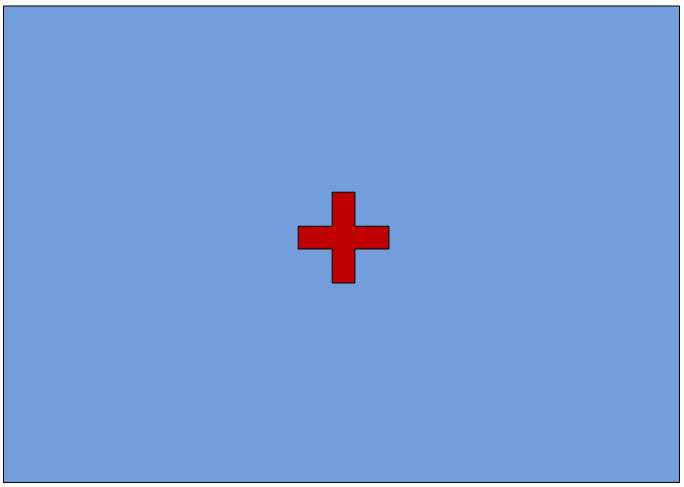

Figura 4.7: Tela usada para o treinamento do estado cerebral durante o desempenho da tarefa fechamento/abertura dos olhos

Foi pedido ao usuário que após o próximo *beep*, ele fechasse os olhos durante os próximos trinta segundos. Esse procedimento em que o usuário deveria intercalar períodos de olhos abertos e de olhos fechados foi realizado seis vezes para cada uma das tarefas desempenhadas. Um total de doze amostras foi coletado. Os primeiros cinco segundos de cada amostra foram rejeitados durante o processamento dos dados para evitar respostas evocadas para o sinal olhos abertos/fechados.

4.4 Processamento dos Sinais

Como a seleção das características do sinal de EEG não foi parte do desenvolvimento deste trabalho, pois o módulo *ThinkGear* já realiza esta etapa do processamento, nesta seção é discutida apenas a realização da classificação das características do sinal de EEG.

As características do sinal de EEG usadas no contexto deste trabalho foram as componentes rítmicas alfa como já foi discutido anteriormente na Seção 3.5.1. As componentes alfa foram traduzidas em sinais que representam o nível de atenção, o fechamento/abertura dos olhos e o desempenho dos movimentos dos braços direito e esquerdo. Para classificação das caracteríticas

alfa foi usado o algoritmo máquina de suporte vetorial (MSV). Este algoritmo foi implementado com o uso da biblioteca LIBSVM [20]. LIBSVM é uma biblioteca para MSV e foi desenvolvida com o objetivo de auxiliar usuários a usar facilmente o algoritmo MSV como uma ferramenta.

As MSV minimizam a probabilidade de se classificar de forma errada padrões ainda não vistos por uma distribuição de probabilidade dos dados fixa e desconhecida. Uma tarefa de classificação geralmente envolve os dados separadamente em conjuntos de treinamento e testes. Cada instância no conjunto de treinamento contém um valor alvo e alguns atributos. O objetivo do algoritmo máquina de suporte vetorial é produzir um modelo baseado nos dados de treinamento, o qual prevê os valores alvo dos dados de teste, sendo dados apenas os atributos dos dados de teste.

Dado um conjunto de treinamento $(x_i, y_i), i = 1, ..., l$, onde $x_i \in \Re^n$ e $y \in \{1, -1\}$, a máquina de suporte vetorial requer uma solução para o seguinte problema de otimização.

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i,$$

sendo,

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i,$$
$$\xi_i \geq 0,$$

onde, $(w, b) \in \Re^n \text{x} \Re^n$ são parâmetros que controlam a função, ξ são os escalares denominados variáveis de folga e C é o peso que o erro exerce na função objetivo.

Os vetores de treinamento x_i são mapeados em um espaço dimensional superior (talvez infinito) pela função ϕ. MSV encontra um hiperplano de separação linear com margem máxima nesse espaço dimensional superior. $K(x_i, x_j) \equiv \phi(x_i^T)\phi(x_j)$ é chamada de função kernel. Os kernels mais usados na literatura são:

- Linear: $K(x_i, x_j) = x_i^T x_j$
- Polinomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- Gaussiano - (RBF): $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$
- *Perceptron* Multi Camadas: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

onde, γ, r e d são parâmetros do kernel.

Para que fosse alcançada uma maior precisão nos resultados obtidos com o uso da biblioteca LIBSVM, os seguintes passos foram realizados:

- Foi realizada a transformação do dado para o formato de um pacote de máquina de suporte vetorial

  O algoritmo máquina de suporte vetorial requer que cada instância do dado seja apresentada como um vetor de números reais.

- Foi realizada uma mudança de escala no dado

  A principal vantagem em realizar uma mudança de escala é evitar que atributos em grandes intervalos numéricos dominem aqueles em pequenos intervalos numéricos. Outra vantagem é evitar dificuldades numéricas durante a realização de cálculos.

- Foi usado o modelo gaussiano para o kernel

  Este kernel não linear foi escolhido porque ele pode manipular o caso em que a relação entre os rótulos de classes e os atributos é não linear, pois ele mapeia amostras em um espaço dimensional superior, diferentemente do kernel linear. A segunda razão para escolha deste kernel foi o número de hiper-parâmetros os quais influenciam na complexidade da seleção do modelo. O kernel polinomial possui mais hiper-parâmetros que o kernel gaussiano. E por último, o kernel gaussiano possui menos dificuldades numéricas.

- Foi usada a validação cruzada para encontrar os melhores parâmetros C e gama

  Existem dois parâmetros em um kernel gaussiano: γ e C. De antemão, não são conhecidos quais C e γ são melhores para um dado problema. Consequentemente, devem ser feitos alguns tipos de modelos de seleção (parâmetros de escolha). O objetivo é identificar bons (C, γ) que podem ser usados para classificar com precisão os dados desconhecidos (dados de teste). A estratégia usada foi primeiramente dividir o conjunto de treinamento em v subconjuntos de mesmo tamanho. Sequencialmente, um subconjunto foi testado usando o classificador treinado a partir dos $v - 1$ subconjuntos restantes.

- Foram usados os melhores parâmetros C e gama para treinar todo o conjunto de treinamento

  Após os melhores (C, γ) serem encontrados, todo o conjunto de treinamento é treinado novamente para gerar o classificador final.

- Foram realizados testes

  Finalmente, os dados de teste foram classificados usando-se o classificador final.

Como o objetivo deste trabalho é fornecer um sistema ICC pervasivo para usuários portadores de deficiência motora severa, o algoritmo MSV usado na classificação das características dos sinais de EEG foi implementado no N810.

4.5 Acesso aos Dispositivos Multimídia

Para acessar os dispositivos multimídia na rede foi usado o arcabouço de *software* BRisa com suporte a UPnP-UP. A versão *python* do BRisa foi usada, pois o arcabouço de *software* desenvolvido no contexto deste trabalho foi implementado em *python*. Para o uso do BRisa foi necessária a instalação do pacote *python-brisa* e algumas dependências.

Um ponto de controle foi desenvolvido para realizar a descoberta de dispositivos multimídia na rede e manipular o conteúdo multimídia do dispositivo selecionado. Esse ponto de controle foi implementado em *PyQt* e por isso foi necessária a instalação do pacote *python-brisa-qtreactor*. O ponto de controle implementado usa as bibliotecas do BRisa para acessar os dispositivos na rede. A tela que apresenta a interface gráfica principal do ponto de controle é mostrada na Figura 4.8.

Figura 4.8: Tela principal do dispositivo de controle

A interface gráfica principal apresenta uma lista com os dispositivos disponibilizados na rede. O ponto de controle envia automaticamente uma mensagem de procura por novos dispositivos na rede. Quando um novo dispositivo é encontrado a lista de dispositivos apresentada ao usuário através da interface gráfica é atualizada.

Para interagir com o ponto de controle, o usuário deve realizar ou imaginar o movimento do braço esquerdo ou do braço direito para percorrer a lista de dispositivos, e fechar os olhos para acessar o dispositivos desejado. Ao acessar um dispositivo uma nova interface gráfica é apresentada ao usuário. A implementação da interface que apresenta o conteúdo multimídia ao usuário depende da aplicação usada pelo usuário. O desenvolvimento de uma aplicação usando o arcabouço de *software*, é discutido na Seção 4.6.6.

Na interface da aplicação multimídia, é mostrado o conteúdo multimídia disponibilizado pelo dispositivo. Esse conteúdo é filtrado devido ao uso do protocolo UPnP-UP que retorna ao usuário apenas os arquivos de sua preferência. Essa filtragem é realizada a partir do perfil

do usuário que contém suas preferências. O perfil do usuário permite descrever informações pessoais do usuário e suas preferências de conteúdo multimídia. Na Figura 4.9 é apresentado um exemplo de perfil do usuário.

```
<?xml version="1.0" encoding="UTF-8">
<up:profile up-id="TBMS1302" xmlns:up="urn:schemas-upnp-up-org:up-1-0">

  <up:personal_data id="personal_profile">
    <up:name>Taciana Saad Rached</up:name>
    <up:username>tacianarached</up:username>
    <!-- Outras informações do perfil ocultadas -->
  </up:personal_data>

  <up:preferences_data>
    <up:container_list>
      <up:container id=0 name=AV>
        <DIDL-Lite xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/"
          xmlns:dc="http://purl.org/dc/elements/1.1/">
          <item>
            <dc:title>telegrama</dc:title>
            <dc:description>A Zeca Baleiro music</dc:description>
            <dc:format>mp3</dc:format>
          </item>
          <!-- Outros itens ocultados -->
        </DIDL-Lite>
      </up:container>
    </up:container_list>
  </up:preferences_data>
</up:profile>
```

Figura 4.9: Listagem de código relativa ao perfil do usuário

4.6 Arcabouço de Software

O arcabouço de *software* foi desenvolvido neste trabalho para auxiliar no desenvolvimento de aplicações ICC pervasivas. Apesar de o arcabouço ter sido desenvolvido com foco nas aplicações de entretenimento ao usuário, ele pode também ser usado em outros tipos de aplicações que podem melhorar a qualidade de vida de uma pessoa portadora de deficiência motora severa. Exemplos dessas aplicações incluem abrir/fechar uma porta, ajustar a temperatura de um ambiente, abrir/fechar cortinas e acender/apagar uma lâmpada, entre outras.

O arcabouço de *software* foi desenvolvido em *PyQt* e possui uma arquitetura baseada em componentes, como foi discutido na Seção 3.7. O arcabouço é composto por alguns módulos *python*. Esses módulos incluem o módulo principal, o controlador de dispositivos, o controlador de sinais classificados, os módulos relativos aos dispositivos multimídia, como o centro de mídias XBMC e os módulos relativos aos sinais classificados como movimento do braço direito, movimento do braço esquerdo, fechamento dos olhos e nível de atenção. Na Figura 4.10 é mostrado o diagrama de blocos que apresenta os módulos presentes no arcabouço.

A implementação do arcabouço foi iniciada com a configuração do ambiente *PyQt* como já foi discutido na Seção 3.7.1. Em seguida, usando a ferramenta Eclipse [5], foi criado um pro-

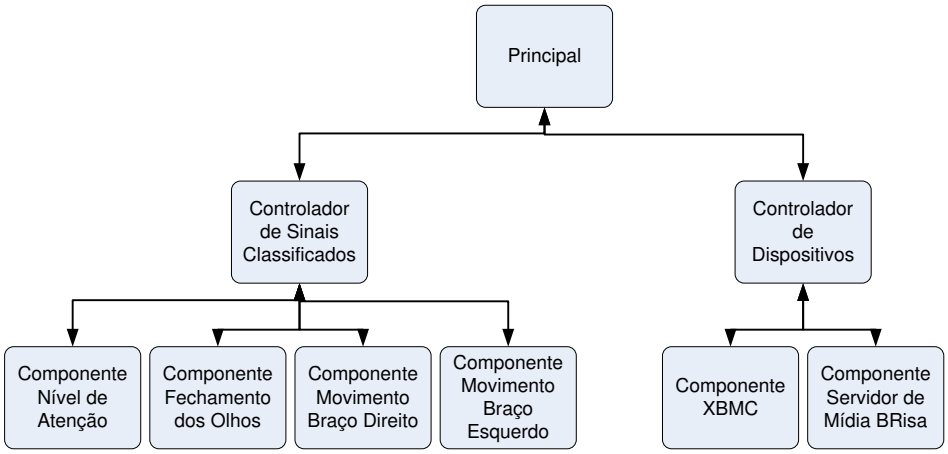

Figura 4.10: Diagrama de blocos dos módulos presentes no arcabouço

jeto *python* para o desenvolvimento dos módulos presentes no arcabouço. Nas próximas seções é discutido o desenvolvimento de cada módulo presente no arcabouço. Para evitar que o texto se torne redundante, será mostrado apenas o processo de desenvolvimento de um componente relativo à um sinal classificado, no caso o referente ao movimento do braço direito, e um componente relativo à um dispositivo multimídia, o centro de mídias XBMC. O processo de desenvolvimento do restante dos componentes relativos aos sinais classificados e dos referentes aos dispositivos multimídia é similar aos apresentados nas Seções 4.6.4 e 4.6.5, respectivamente.

4.6.1 Desenvolvimento do Módulo Principal

O módulo principal foi implementado com a função de realizar a comunicação entre a interface gráfica da aplicação e o restante do arcabouço. No desenvolvimento deste módulo, primeiramente foram importadas as bibliotecas necessárias para o seu desenvolvimento. Essas bibliotecas incluem *QtCore* e *QtGui* de *PyQt*, as bibliotecas *sys* e *socket* de *python*, o módulo *controladorDeSinais* e o módulo *controladorDeDispositivos*. Na Figura 4.11 é apresentada a listagem com o código usado para importar as bibliotecas citadas anteriormente.

```
import socket
import sys

from PyQt4.QtCore import *
from PyQt4.QtGui import *

import controladorDeSnais
import controladorDeDispositivos
```

Figura 4.11: Listagem de código relativa as bibliotecas importadas no módulo principal

Depois de importar as bibliotecas, o primeiro passo foi implementar uma classe, chamada

Principal, que herda os métodos da classe *QThread* do módulo *QtCore*. Foi necessário o uso da classe *QThread* para permitir que a interface gráfica da aplicação permanecesse executando enquanto a aplicação aguarda a chegada dos sinais classificados. A classe *Principal* recebe como parâmetro o dispositivo multimídia selecionado pelo usuário do sistema. No método usado para inicialização das variáveis, a classe *Principal* é iniciada com o uso de *super()* e os controladores de sinais classificados e de dispositivos são instanciados. Além disso, o módulo *socket* também é instanciado e o método usado para conectar o *socket* ao servidor que disponibiliza os sinais classificados é executado. A listagem de código relativa à implementação da classe e sua inicialização é apresentada na Figura 4.12.

```
class Principal(QThread):
    def __init__(self, dispositivo, parent=None):
        super(Principal, self).__init__(parent)
        self.controlador = \
            controladorDeSnais.ControladorSinaisClassificados(dispositivo)
        self.controlador2 = \
            controladorDeDispositivos.ControladorDispositivos(dispositivo)

        try:
            self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        except socket.error, msg:
            sys.stderr.write("[ERROR] %s\n" % msg[1])
            sys.exit(1)
        self.conectarSocket()
```

Figura 4.12: Listagem de código relativa a implementação e inicialização da classe *Principal*

A conexão entre a classe *Principal* e o servidor é realizada através do método *connect()* da classe *socket* que é executado pelo método *conectarSocket()* da classe *Principal*. Na Figura 4.13 é mostrada a listagem de código que apresenta o método implementado para conectar a classe *Principal* ao servidor.

```
def conectarSocket(self):
    try:
        self.sock.connect(("127.0.0.1", 13854))
    except socket.error, msg:
        sys.stderr.write("[ERROR] %s\n" % msg[1])
        sys.exit(2)
```

Figura 4.13: Listagem de código relativa a implementação do método *conectarSocket()*

O próximo método implementado foi o *run()* que é executado quando a *thread* da classe Principal é iniciada. Esse método recebe os sinais classificados quando estes são disponibilizados pelo servidor. Quando um sinal classificado é recebido, o método *recebeSinalClassificado()* é executado. O método *run()* é mostrado na Figura 4.14.

```
def run(self):
    sinal = self.sock.recv(1024)
    while len(sinal):
        sinal = self.sock.recv(1024)
        self.recebeSinalClassificado(sinal)
```

Figura 4.14: Listagem de código relativa a implementação do método *run()*

O método *recebeSinalClassificado()* tem como função enviar o sinal classificado ao controlador de sinais classificados. Esse método recebe como parâmetro o sinal classificado e o transmite ao controlador através do método *recebeSinalClassificado()* do controlador. A listagem de código relativa à implementação do método *recebeSinalClassificado()* é apresentada na Figura 4.15.

```
def recebeSinalClassificado(self, sinal_classificado):
    self.sinal_classificado = sinal_classificado
    self.controlador.recebeSinalClassificado(self.sinal_classificado)
    self.recebeEvento()
```

Figura 4.15: Listagem de código relativa a implementação do método *recebeSinalClassificado()*

O método *recebeEvento()* tem a função de receber o evento para o dispositivo multimídia selecionado e o enviar para o controlador de dispositivos. Para que essa função seja realizada, esse método executa o método *retornaEvento()* da classe *ControladorSinaisClassificados*. O método retornaEvento() retorna o evento que é passado como parâmetro para o método *enviaEvento()*. A listagem de código relativa a implementação do método *recebeEvento()* é mostrada na Figura 4.16.

```
def recebeEvento(self):
    self.evento = self.controlador.retornaEvento()
    self.enviaEvento(self.evento)
```

Figura 4.16: Listagem de código relativa a implementação do método *recebeEvento()*

O método *enviaEvento()* envia o evento para o controlador de dispositivos através do método *recebeEvento()* da classe *ControladorDispositivos* e em seguida executa o método *recebeComando()*. O método *recebeComando()* recebe o comando a ser usado para controlar o dispositivo multimídia e envia esse comando para interface gráfica. O envio desse comando para interface é realizado através de um mecanismo de *PyQt* chamado sinais e *slots*. Quando um comando é recebido, um sinal é emitido para interface para que a mesma seja atualizada. A listagem de código relativa aos métodos *enviaEvento()* e *recebeComando()* é apresentada na

Figura 4.17.

```
def enviaEvento(self, comando):
    self.controlador2.recebeEvento(self.comando)
    self.recebeComando()

def recebeComando(self):
    self.comando = self.controlador2.retornaComando()
    self.emit(SIGNAL("indexed(QString)"), self.comando)
```

Figura 4.17: Listagem de código relativa a implementação dos métodos *enviaEvento()* e *recebeComando()*

4.6.2 Desenvolvimento do Controlador de Sinais Classificados

O controlador de sinais classificados tem a função de gerenciar qual componente deve ser carregado quando ocorrer a chegada de um sinal classificado. Esse controlador é usado como uma ponte entre o módulo *principal* e o componente relativo ao sinal classificado recebido. Para implementação do controlador de sinais classificados, primeiramente foi criado um módulo *python*. Nesse módulo, foram importados os componentes relativos aos sinais classificados como movimento do braço direito, movimento do braço esquerdo, fechamento dos olhos e nível de atenção. Na Figura 4.18 é apresentada a listagem de código relativa aos componentes importados no módulo *controladorDeSinais*.

```
import bracoDireito
import bracoEsquerdo
import fechamentoOlhos
import nivelAtencao
```

Figura 4.18: Listagem de código relativa aos componentes importados no módulo *controladorDeSinais*

Depois de importar os componentes, a classe *ControladorSinaisClassificados* foi implementada. Essa classe recebe como parâmetro o dispositivo selecionado pelo usuário do sistema. O método de inicialização da classe é usado para setar o dispositivo. A listagem de código relativa a implementação da classe e o método de inicialização é apresentada na Figura 4.19.

O próximo passo no desenvolvimento do controlador de sinais classificados foi a implementação do método usado para receber o sinal classificado. O método *recebeSinalClassificado()* recebe como parâmetro o sinal classificado e o passa para o método *carregaComponente()*. A listagem de código relativa a implementação do método *recebeSinalClassificado* é apresentada na Figura 4.20.

```
class ControladorSinaisClassificados:
    def __init__(self, dispositivo, parent=None):
        self.dispositivo = dispositivo
```

Figura 4.19: Listagem de código relativa a implementação e inicialização da classe *ControladorSinaisClassificados*

```
def recebeSinalClassificado(self, sinal_classificado):
    self.sinal_classificado = sinal_classificado
    self.carregaComponente(self.sinal_classificado)
```

Figura 4.20: Listagem de código relativa a implementação do método *recebeSinalClassificado()*

O método *carregaComponente()* é responsável por ativar o componente relativo ao sinal classificado recebido. Esse método recebe como parâmetro o sinal classificado e instancia o componente relativo a este sinal. A listagem de código relativa a implementação do método *carregaComponente()* é mostrada na Figura 4.21.

```
def carregaComponente(self, sinal_classificado):
    if self.sinal_classificado != None:

        if self.sinal_classificado == "braco_direito":
            self.componente = bracoDireito.BracoDireito(self.dispositivo)

        elif self.sinal_classificado == "braco_esquerdo":
            self.componente = bracoEsquerdo.BracoEsquerdo(self.dispositivo)

        elif self.sinal_classificado == "fechamento_olhos":
            self.componente = fechamentoOlhos.FechamentoOlhos(self.dispositivo)

        elif self.sinal_classificado == "nivel_atencao":
            self.componente = nivelAtencao.NivelAtencao(self.dispositivo)
```

Figura 4.21: Listagem de código relativa a implementação do método *carregaComponente()*

O método *retornaEvento()* recebe o evento para o dispositivo multimídia selecionado e o envia para a classe *Principal*. O evento é recebido através do método *retornaEvento()* do componente relativo ao sinal classificado recebido. Após retornar o evento para a classe *Principal*, o componente é descarregado. Isso acontece quando o método *descarregaComponente()* é executado. A listagem de código relativa aos métodos *retornaEvento()* e *descarregaComponente()* é mostrada na Figura 4.22.

```
def retornaEvento(self):
    self.evento = self.componente.retornaEvento()
    return self.evento
    self.descarregaComponente()

def descarregaComponente(self):
    self.sinal_classificado = None
    self.componente = None
```

Figura 4.22: Listagem de código relativa a implementação dos métodos *retornaEvento* e *descarregaComponente()*

4.6.3 Desenvolvimento do Controlador de Dispositivos Multimídia

O controlador de dispositivos é usado para gerenciar o carregamento do componente relativo ao dispositivo multimídia selecionado pelo usuário do sistema. Esse controlador é usado como uma ponte entre o módulo principal e o componente relativo ao dispositivo selecionado. Para o desenvolvimento desse controlador foi criado um módulo *python*, onde foram importados os componentes referentes aos dispositivos multimídia. A listagem de código relativo aos componentes importados pelo módulo *controladorDeDispositivos* é apresentada na Figura 4.23.

```
import xbmc
import brisaServer
```

Figura 4.23: Listagem de código relativa aos componentes importados no módulo *controladorDeDispositivos*

Depois de importar os componentes, a classe *ControladorDispositivos* foi implementada. Essa classe recebe como parâmetro o dispositivo selecionado pelo usuário do sistema. O método de inicialização da classe é usado para setar o dispositivo e executar o método *carregaComponente()*. A listagem de código relativa a implementação da classe e o método de inicialização é apresentada na Figura 4.24.

```
class ControladorDispositivos:
    def __init__(self, dispositivo):
        self.dispositivo = dispositivo
        self.carregaComponente()
```

Figura 4.24: Listagem de código relativa a implementação e inicialização da classe *ControladorDispositivos*

O método *carregaComponente()* ativa o componente relativo ao dispositivo multimídia selecionado. Esse método instancia o componente relativo ao dispositivo. A listagem de código

relativa a implementação do método *carregaComponente()* é mostrada na Figura 4.25.

```
def carregaComponente(self):
    if self.dispositivo == "XBMC":
        self.componente = xbmc.XBMC()

    elif self.dispositivo == "BRisa_Server":
        self.componente = brisaServer.BRisaServer()
```

Figura 4.25: Listagem de código relativa a implementação do método carregaComponente()

Foram desenvolvidos um método para receber o evento mapeado pelo componente do sinal classificado chamado *recebeEvento()* e um método para enviar esse evento ao componente relativo ao dispositivo selecionado, chamado *enviaEvento()*.A listagem de código relativa a implementação dos métodos *recebeEvento()* e *enviaEvento()* é mostrada na Figura 4.26.

```
def recebeEvento(self, evento):
    self.evento = evento
    self.enviaEvento(self.evento)

def enviaEvento(self, evento):
    self.comando = self.componente.executaComando(evento)
    self.retornaComando(self.comando)
```

Figura 4.26: Listagem de código relativa a implementação dos métodos *recebeEvento()* e *enviaEvento()*

Por último, foi implementado um método chamado *retornaComando()* com a função de enviar o comando transmitido pelo componente ao módulo principal. Esse comando é usado na atualização da interface gráfica. A listagem de código relativa a implementação do método *retornaComando()* é mostrada na Figura 4.27.

```
def retornaComando(self, comando):
    return comando
```

Figura 4.27: Listagem de código relativa a implementação do método *retornaComando()*

4.6.4 Desenvolvimento do Componente Relativo ao Sinal Classificado como Movimento do Braço Direito

O desenvolvimento do componente relativo ao sinal classificado como movimento do braço direito é apresentado para ilustrar os passos discutidos na Seção 3.7.1 para implementação deste

tipo de componente. Esse componente é carregado pelo controlador de sinais classificados sempre que o usuário do sistema realiza ou imagina o movimento do braço direito. Para iniciar a discussão sobre a implementação desse componente, é suposto que o ambiente de desenvolvimento *PyQt* já estivesse configurado.

O primeiro passo no desenvolvimento de um componente relativo à um sinal classificado é a criação de um módulo *python*. No exemplo em questão, esse módulo é chamado *bracoDireito*. O segundo passo é a implementação de uma classe referente ao sinal classificado que receba como parâmetro o dispositivo selecionado pelo usuário do sistema. Nesse caso a classe implementada é denominada *BracoDireito*. O método de inicialização é usado para setar o dispositivo selecionado. A listagem de código relativa a implementação da classe e o método de inicialização é apresentada na Figura 4.28.

```
class BracoDireito:
    def __init__(self, dispositivo, parent = None):
        self.dispositivo = dispositivo
```

Figura 4.28: Listagem de código relativa a implementação e inicialização da classe *BracoDireito*

O segundo passo é a implementação do método usado para mapear o sinal classificado em um evento para o dispositivo multimídia, chamado *defineEvento()*. Esse método contém uma condição para cada componente relativo a um dispositivo multimídia. Como no arcabouço existem dois componentes relativos aos dispositivos multimídia, o centro de mídias XBMC e o servidor do BRisa, no método *defineEvento()* deve existir uma condição para cada um desses dois dispositivos. A listagem de código relativa a implementação do método *defineEvento()* é mostrada na Figura 4.29.

```
def defineEvento(self):
    if self.dispositivo == "XBMC":
        self.evento = "playNext"

    elif self.dispositivo == "BRisa_Server":
        self.evento = "selectNextDirectory"
```

Figura 4.29: Listagem de código relativa a implementação do método *defineEvento()*

O terceiro passo é o desenvolvimento de um método com a função de enviar o evento ao controlador de sinais classificados, denominado *retornaEvento()*. A listagem de código relativa a implementação do método *retornaEvento()* é mostrada na Figura 4.30.

O quarto passo é importar o componente desenvolvido, no módulo *controladorDeSinais* para realizar a comunicação entre o componente e o restante do arcabouço. A listagem de

```
def retornaEvento(self):
    self.defineComando()
    return self.evento
```

Figura 4.30: Listagem de código relativa a implementação do método *retornaEvento()*

código relativo ao componente importado pelo módulo *controladorDeSinais* é apresentada na Figura 4.31.

```
import bracoDireito
```

Figura 4.31: Listagem de código relativa ao componente importado no módulo *controladorDeSinais*

O quinto passo é a implementação de uma condição para o novo componente no método *carregaComponente()* do controlador de sinais. E o último passo é instanciar o componente na condição implementada. Os dois últimos passos são ilustrados na listagem de código apresentada na Figura 4.32.

```
def carregaComponente(self, sinal_classificado):
    if self.sinal_classificado != None:

        if self.sinal_classificado == "braco_direito":
            self.componente = bracoDireito.BracoDireito(self.dispositivo)
```

Figura 4.32: Listagem de código relativa a implementação de uma condição para o novo componente no módulo *controladorDeSinais*

4.6.5 Desenvolvimento do Componente Relativo ao Dispositivo Multimídia XBMC

O desenvolvimento do componente relativo ao dispositivo centro de mídias XBMC é apresentado para ilustrar os passos discutidos na Seção 3.7.1 para implementação deste tipo de componente. Esse componente é carregado pelo controlador de dispositivos quando o usuário do sistema deseja controlar o centro de mídias XBMC. Para iniciar a discussão sobre a implementação desse componente, é suposto que o ambiente de desenvolvimento *PyQt* já estivesse configurado.

O primeiro passo no desenvolvimento de um componente relativo à um dispositivo é a criação de um módulo *python*. No exemplo em questão esse módulo é chamado *xbmc*. O segundo

passo é a implementação de uma classe referente ao dispositivo selecionado. Nesse caso a classe implementada é denominada *XBMC*.

O segundo passo é a implementação do método usado para mapear o evento recebido em um comando para o dispositivo multimídia e para a interface gráfica, chamado *executaComando()*. Esse método recebe como parâmetro um evento para o dispositivo e contém uma condição para cada tipo de evento relacionado a um comando. A listagem de código relativa a implementação do método *executaComando()* é mostrada na Figura 4.33.

```
def executaComando(self, evento):
    if evento == "play":
        self.play()

    elif evento == "pause":
        self.pause()

    elif evento == "stop":
        self.stop()

    elif evento == "playNext":
        self.playNex()

    elif evento == "back":
        self.playPrev()

    elif evento == "exit":
        self.exit()

    return evento
```

Figura 4.33: Listagem de código relativa a implementação do método *executaComando()*

O terceiro passo é o desenvolvimento de um método para cada comando a ser usado no controle do dispositivo multimídia. A listagem de código relativa a implementação dos métodos usados no controle do dispositivo XBMC é mostrada na Figura 4.34.

O quarto passo é importar o componente desenvolvido, no módulo *controladorDeDispositivos* para realizar a comunicação entre o componente e o restante do arcabouço. A listagem de código relativa ao componente importado pelo módulo *controladorDeDispositivos* é apresentada na Figura 4.35.

O quinto passo é a implementação de uma condição para o novo componente no método *carregaComponente()* do controlador de dispositivos. E o último passo é instanciar o componente na condição implementada. Os dois últimos passos são ilustrados na listagem de código apresentada na Figura 4.36.

```
def play(self):
    urllib.urlopen("http://192.168.1.187/xbmcCmds/xbmcHttp?" \
                   "command=PlayFile(F:\music\test.mp3)")

def pause(self):
    urllib.urlopen("http://192.168.1.187/xbmcCmds/xbmcHttp?command=Pause()")

def playNex(self):
    urllib.urlopen("http://192.168.1.187/xbmcCmds/xbmcHttp?command=PlayNext()")

def playPrev(self):
    urllib.urlopen("http://192.168.1.187/xbmcCmds/xbmcHttp?command=PlayPrev()")

def stop(self):
    urllib.urlopen("http://192.168.1.187/xbmcCmds/xbmcHttp?command=Stop()")

def exit(self):
    urllib.urlopen("http://192.168.1.187/xbmcCmds/xbmcHttp?command=Exit()")
```

Figura 4.34: Listagem de código relativa a implementação dos métodos usados no controle do centro de mídias XBMC

```
import xbmc
```

Figura 4.35: Listagem de código relativa ao componente importado no módulo *controladorDeDispositivos*

4.6.6 Desenvolvimento de Aplicação

Para exemplificar o uso do arcabouço, foi desenvolvida uma aplicação para o controle do centro de mídias XBMC. Essa aplicação foi implementada seguindo os passos discutidos na Seção 3.7.2. Primeiramente o ambiente de desenvolvimento *PyQt* foi configurado para a plataforma *Maemo*. Para instalação da versão *PyQt4* para *Maemo* foram realizados os seguintes passos:

- Foi habilitado o repositório de *extras-devel*.
- Foi executado o commando *apt-cache search python2.5-qt4*.
- Foi executado o comando *apt-get install python2.5-qt4-core python2.5-qt4-gui*.

Depois de instalar os pacotes necessários para a configuração do ambiente, foi realizado o *download* do arcabouço. Como o dispositivo escolhido é o centro de mídias XBMC, não foi necessário o desenvolvimento de nenhum componente relativo ao dispositivo, pois o arcabouço já contém o componente relativo a esse centro. Também não foi necessário o desenvolvimento de nenhum componente relativo aos sinais classificados, pois os componentes necessários para a implementação desta aplicação já existem no arcabouço.

```
def carregaComponente(self):
    if self.dispositivo == "XBMC":
        self.componente = xbmc.XBMC()
```

Figura 4.36: Listagem de código relativa a implementação de uma condição para o novo componente no módulo *controladorDeDispositivos*

O próximo passo foi o desenvolvimento do dispositivo de controle. Para a implementação do dispositivo de controle foi necessária a instalação do BRisa e suas dependências. Esse dispositivo foi desenvolvido para acessar os dispositivos multimídia disponibilizados na rede como discutido na Seção 4.5. A tela da interface gráfica principal do dispositivo de controle desenvolvido, foi mostrada na Figura 4.8, apresentada anteriormente.

Em seguida foi desenvolvida a interface gráfica para realimentação do usuário. Na parte inferior da interface são apresentados os botões referentes aos comandos usados no controle do dispositivo XBMC. Do lado esquerdo é mostrada uma legenda que ilustra quais tarefas devem ser desempenhadas pelo usuário do sistema para que determinado comando seja executado. E no lado direito são apresentadas as músicas preferidas do usuário de acordo com o seu perfil. A tela da aplicação usada no controle do centro de mídias XBMC é apresentada na Figura 4.37.

Figura 4.37: Tela da aplicação desenvolvida para controlar o centro de mídias XBMC usada para realimentar o usuário do sistema

No módulo onde foi desenvolvida a interface gráfica, foi importado o módulo principal do arcabouço. Além disso, na classe *GUI* a *thread* da classe *Principal* foi iniciada para permitir que a interface aguardasse sinais de atualização enquanto o arcabouço recebe os sinais classificados. Foi implementado também na classe *GUI* o mecanismo de sinais e *slots* de *PyQt* para que fossem recebidos os sinais de atualização da interface.

4.7 Teste de Usabilidade

Nesta Seção, são discutidas as etapas de planejamento e elaboração dos ensaios de usabilidade a serem realizados para avaliação do sistema ICC pervasivo para o controle de dispositivos multimídia. O planejamento e a elaboração do teste foram realizados de acordo com o protocolo experimental desenvolvido no LIHM e discutido na Seção 3.9. Após a classificação do sistema quanto a fase de desenvolvimento em que ele se encontra, sua natureza e a natureza do teste, usou-se o guia para aplicação do protocolo para seleção de quais itens deveriam constar nos artefatos a serem gerados para a aplicação da avaliação. Os artefatos gerados incluem o plano geral do ensaio de avaliação, o plano de avaliação com usuário, o plano de coleta de dados, o roteiro de tarefa de testes, a ficha de observação, questionário pré-teste, questionário pós-teste, ficha de cadastro e o termo de aceitação das condições de realização do teste. Todos os artefatos podem ser encontrados no Apêndice A.

Nos ensaios de usabilidade para a avaliação do sistema desenvolvido no contexto deste trabalho, tem-se o objetivo de detectar os fatores que colaboram para elevar o nível de dificuldade no uso do sistema, a ocorrência de erros, insatisfação e desconforto do usuário, avaliar a usabilidade e a mobilidade do sistema. O levantamento dessas informações deve ser feito a partir da observação da interação entre o usuário e o sistema. O tempo disponível para realização dos testes é de sete semanas e os recursos disponíveis são os encontrados no Laboratório LIHM, além de um notebook, câmeras portáteis, microfone de lapela, equipamento de EEG, dispositivo móvel N810 e aplicação multimídia ICC pervasiva. A partir da realização dos testes, espera-se obter como resultados a identificação do nível de dificuldade e satisfação do usuário no uso do sistema. Além disso, espera-se identificar as falhas, os problemas de usabilidade e receber sugestões de melhorias para o *software* e verificar a mobilidade do *hardware*.

O sistema ICC pervasivo para o controle de dispositivos multimídia habilita usuários portadores de deficiência motora severa a interagir com dispositivos multimídia que estejam inseridos em seus ambientes. O controle dos dispositivos é realizado através da atividade cerebral do usuário. Esse sistema consiste de um equipamento de EEG para aquisição e parte do processamento dos sinais cerebrais e um dispositivo móvel responsável pela classificação dos sinais, execução do arcabouço de software, acesso aos dispositivos multimídia, personalização de serviços de acordo com o perfil do usuário e execução de aplicação multimídia. No uso do sistema, os eletrodos do equipamento de EEG são colocados no escalpo (couro cabeludo) do usuário para monitoramento de seus sinais cerebrais. O usuário deve realizar tarefas pré-definidas para gerar um determinado estado cerebral que é usado para controlar dispositivos multimídia, ou seja, as intenções do usuário são traduzidas em sinais de comando para os dispositivos. Esse sistema pode ser usado em qualquer lugar e a qualquer hora onde exista uma rede local ou internet. Para que um usuário possa operá-lo é necessária a realização de uma etapa de treinamento. Uma

discussão sobre o treinamento realizado com os usuários é apresentada na Seção 4.3.

Após receber o treinamento necessário para aprender como interagir com o sistema, o usuário deve desempenhar tarefas de manipulação de conteúdo multimídia com arquivos de música, a partir de seus sinais cerebrais, que são enviados ao dispositivo móvel que contém o software responsável pela classificação e tradução desses sinais em sinais de comando para o dispositivo de mídia. Essas tarefas são realizadas individualmente pelo usuário do sistema e para que elas possam ser desempenhadas com sucesso é necessário que um dispositivo móvel esteja conectado a rede local ou internet, para que seja feita uma busca por dispositivos multimídia.

Os participantes dos ensaios como usuários, devem ser trinta pessoas saudáveis para os testes realizados em laboratório e dez pessoas portadoras de deficiência motora severa para os testes feitos em campo. Não é desejada a realização de testes com pessoas portadoras de deficiência motora leve, que possuem o movimento dos braços. Todos os usuários devem ser inexperientes no uso do sistema. No recrutamento dos usuários, os meios de contato a serem usados são telefone e *email*. Os usuários a serem recrutados devem possuir o perfil descrito anteriormente e devem ser informados que o propósito do ensaio é a avaliação da usabilidade de um produto para controle de dispositivos multimídia através da atividade cerebral por usuários portadores de deficiência motora severa. O contato com o usuários selecionados dever ser feito por telefone ou *email*.

O objetivo geral da avaliação é investigar o processo interativo de usuários com o sistema a fim de observar quais são os aspectos que mais dificultam o uso do sistema. A avaliação possui natureza qualitativa, quantitativa, objetiva, subjetiva e somativa. A análise qualitativa a ser realizada é baseada na observação do usuário, na aplicação de um questionário pós-teste e na realização de uma entrevista ao fim do ensaio. A linguagem corporal e a verbalização durante a realização de uma tarefa são umas das reações do usuário a serem observadas durante a realização do teste. O questionário a ser aplicado após o teste aborda questões a respeito da usabilidade do sistema. A entrevista a ser realizada ao fim do ensaio deve ser contextualizada com relação ao uso do sistema pelo usuário.

A avaliação quantitativa a ser realizada é baseada em um questionário aplicado após o teste. Esse questionário aborda questões a respeito da usabilidade do sistema e suas questões contêm escalas para medir o nível de satisfação do usuário. A análise objetiva a ser realizada é baseada em aspectos contáveis a serem observados pelo avaliador, como o número de erros que o usuário cometeu durante a realização de uma tarefa, quantas tarefas o usuário conseguiu finalizar com sucesso ou quantas tarefas o usuário não conseguiu finalizar, quanto tempo o usuário gastou para realizar uma tarefa, entre outros.

A avaliação subjetiva a ser realizada aborda aspectos com relação a opinião do usuário sobre o sistema. Exemplos disso incluem o que o usuário achou do sistema, se ele achou fácil

ou difícil de usar, se ele achou o sistema atrativo, entre outros. Essa avaliação é realizada através da aplicação de questionário e entrevista. Na análise somativa a ser realizada tem-se o objetivo de identificar os problemas existentes no sistema, adequar a interface e investigar possíveis soluções para os problemas de usabilidade identificados.

O tempo estimado para realização da avaliação é de uma hora e cinquenta minutos, onde 1 hora é o tempo gasto no treinamento, vinte minutos no pré-teste, vinte minutos no teste e dez minutos no pós-teste.

Os métodos a serem usados na avaliação dos quarenta usuários são a observação direta, o perfil e a satisfação do usuário. São quarenta sessões de testes a serem realizadas, sendo trinta delas com usuários saudáveis no LIHM-UFCG e dez delas com pessoas portadoras de deficiência motora severa em campo no Instituto de Neurociências da Paraíba. Para participar do ensaio como avaliador é necessário saber manipular um equipamento de EEG.

Apesar dos ensaios de usabilidade ainda não terem sido realizados de maneira adequada como discutido nesta Seção, foram realizados alguns testes em ambiente não controlado com dois usuários para verificar o funcionamento do sistema. As tarefas realizadas pelos usuários nesses testes foram as discutidas na elaboração do material para a avaliação de usabilidade. Pode-se observar a partir destes testes, que apesar de ser um pouco penoso para o usuário o uso desse sistema, devido a necessidade de realização de um treinamento, é possível a realização do controle de dispositivos a partir das intenções do usuário.

4.8 Sumário

Neste capítulo foi apresentado o desenvolvimento do sistema proposto neste trabalho. A implementação do sistema foi dividida em etapas: *hardware*, aquisição dos sinais, treinamento, processamento, acesso aos dispositivos multimídia e arcabouço. O *hardware* foi desenvolvido para adquirir os sinais cerebrais a partir de quatro canais e realizar uma parte do processamento do sinal. O *hardware* consiste de eletrodos de Ag/AgCl, quatro chips *ThinkGear*, um microcontrolador *Arduino Nano* e um módulo *Bluetooth* acoplados a um boné. O módulo *Bluetooth* é responsável pela comunicação entre os componentes citados e o restante do sistema que foi implementado no N810.

A aquisição dos sinais cerebrais do usuário do sistema foi realizada com o uso do equipamento desenvolvido neste trabalho. Os eletrodos do *hardware* foram aplicados a superfície do couro cabeludo do usuário para obtenção de sua atividade cerebral. Para melhorar a condutividade dos sinais foi usada uma pasta condutora nos eletrodos. Foi realizado um treinamento com os usuários para que eles aprendessem a operar o sistema adequadamente. Durante o treinamento o usuário foi instruído a realizar determinadas tarefas e a manter um determinado estado cerebral durante a execução dessas tarefas.

A seleção das características do sinal foi realizada no N810 com o uso da biblioteca LIB-SVM. Essa biblioteca implementa o algoritmo de aprendizagem máquinas de suporte vetorial. O arcabouço BRisa foi usados para o acesso aos dispositivos multimídia. Foi desenvolvido um ponto de controle que implementa os métodos do ponto de controle do BRisa para realizar a descoberta automática de dispositivos na rede. A versão do BRisa usada foi a que possui suporte ao protocolo UPnP-UP para que os serviços disponibilizados ao usuário fossem personalizados.

Foi desenvolvido um arcabouço de *software* para facilitar a implementação de aplicações ICC pervasivas. Esse arcabouço possui uma arquitetura baseada em componentes. O arcabouço é composto de dois controladores: de sinais classificados e de dispositivos. No controlador de sinais classificados são incluídos quatro componentes e o controlador de dispositivos por dois componentes, como foi discutido na Seção 4.6.

Por último, foram discutidos o planejamento e a elaboração do teste de usabilidade a ser realizado para avaliar o sistema desenvolvido neste trabalho. O processo de planejamento e elaboração foi realizado a partir do protocolo experimental desenvolvido no Laboratório de Interface Homem máquina da Universidade Federal de Campina Grande.

Capítulo 5

Conclusões e Trabalhos Futuros

A inserção de pessoas portadoras de deficiência na sociedade vem sendo foco de diversas pesquisas com o objetivo de promover uma melhora na qualidade de vida delas. O desenvolvimento de tecnologias assistivas busca habilitar essas pessoas a desempenhar funções que antes não eram possíveis. Nesse contexto, sistemas ICC permitem que pessoas com deficiência motora severa controlem dispositivos disponibilizados no meio no qual elas estão inseridas.

A integração dos sistemas ICC, dispositivos móveis e serviços pervasivos, além de permitir a interação entre os indivíduos citados anteriormente e os dispositivos encontrados em seus meios, também fornecem mais mobilidade ao usuário do sistema. Sistemas ICC pervasivos habilitam usuários a interagir com os dispositivos citados em qualquer lugar, a qualquer hora.

Neste trabalho apresentou-se uma arquitetura para o desenvolvimento de sistemas ICC pervasivos para o controle de dispositivos multimídia com o objetivo de permitir a interação entre pessoas com deficiência motora severa e os dispositivos citados e com isso melhor qualidade de vida dessas pessoas. O principal foco na definição e implementação da arquitetura introduzida foi fornecer um arcabouço de *software* para facilitar o desenvolvimento de aplicações ICC para interação com conteúdo multimídia.

Além disso, discutiu-se o desenvolvimento de aplicações simples que proporcionam aos usuários mais mobilidade, do ponto de vista do dispositivo, e independência, permitindo que pessoas portadoras de deficiência motora severa controlem dispositivos multimídia em qualquer lugar e a qualquer hora. Um sistema para o controle do centro de mídias XBMC foi desenvolvido para validar a arquitetura introduzida. O sistema desenvolvido permite que usuários executem músicas e vídeos utilizando informações sobre sua atividade cerebral. Para isso, foi necessário no desenvolvimento do sistema, a implementação de um equipamento de EEG para aquisição da atividade cerebral, do algoritmo para classificação dos sinais, do arcabouço de software, de um mecanismo para o acesso aos dispositivos multimídia, personalização de serviços de acordo com o perfil do usuário e da aplicação multimídia em um dispositivo móvel. No contexto deste trabalho, também foram realizadas as etapas de planejamento e elaboração do

ensaio de usabilidade a ser realizado para avaliação do sistema desenvolvido.

O *hardware* desenvolvido para a aquisição dos sinais cerebrais além de adquirir os sinais de EEG, ele também executa o pré-processamento do sinal, amplificação e digitalização, e a seleção das características relevantes do sinal. Esse equipamento possui quatro canais, e seus eletrodos e o sistema embarcado, que realiza parte do processamento do sinal, são inseridos em um boné, o que facilita o uso do *hardware* pelo usuário do sistema. O restante do sistema é composto pelo dispositivo móvel N810. Para classificação do sinal foi implementado nesse dispositivo o algoritmo MSV, descrito no Capítulo 4. Além disso, o N810 foi usado para acessar os dispositivos multimídia disponibilizados na rede com o uso do arcabouço BRisa. O BRisa realiza a descoberta automática de dispositivos em uma rede local ou Internet e permite a manipulação de conteúdo multimídia. Além disso, o BRisa com suporte ao protocolo UPnP-UP foi usado para a personalização de serviços de acordo com o perfil do usuário do sistema. O arcabouço de *software* desenvolvido no contexto deste trabalho possui uma arquitetura baseada em componentes e é usado para auxiliar no desenvolvimento de aplicações multimídia ICC.

Apesar de ainda não terem sido realizadas as avaliações de usabilidade, foram feitos alguns testes com o sistema ICC pervasivo desenvolvido neste trabalho, em ambiente não controlado, com usuários saudáveis. Os resultados desses testes mostraram que é possível inserir pessoas portadoras de doenças neuromusculares severas na sociedade, com o desenvolvimento de aplicações ICC pervasivas.

Para obter resultados mais significativos a respeito do sistema desenvolvido neste trabalho com relação a sua usabilidade, falhas e mobilidade, deve ser realizado o ensaio de usabilidade planejado e elaborado para avaliação do sistema. Com o objetivo de explorar novas abordagens a serem aplicadas junto com ICC, é relevante em trabalhos futuros, investigar o uso de informações a respeito da localização do usuário para auxiliar na seleção do dispositivo multimídia que o usuário deseja controlar. Além disso, deve-se investigar caminhos de interação mais complexos, usando mais eletrodos para aquisição dos sinais de EEG em diferentes regiões do cérebro e outros mecanismos de interação, como a Eletrooculografia (EOG) [18, 19].

Referências Bibliograficas

[1] Amyotrophic lateral sclerosis. http://www.alsa.org/. [Online; accessed 04-April-2010].

[2] Biblioteca newsoftserial. http://arduiniana.org/libraries/NewSoftSerial. [Online; accessed 11-May-2010].

[3] Brisa. http://brisa.garage.maemo.org/. [Online; accessed 20-May-2010].

[4] Cerebral palsy. http://www.cerebralpalsy.org/. [Online; accessed 13-April-2010].

[5] Eclipse. http://www.eclipse.org/. [Online; accessed 20-May-2010].

[6] Media center xbmc. http://xbmc.org/. [Online; accessed 04-May-2010].

[7] Microcontrolador arduino nano. http://www.arduino.cc/. [Online; accessed 11-May-2010].

[8] Multiple sclerosis. http://www.nationalmssociety.org/. [Online; accessed 13-April-2010].

[9] NeuroSky. http://www.neurosky.com. [Online; accessed 04-April-2010].

[10] Pyqt. http://www.riverbankcomputing.co.uk/software/pyqt/intro. [Online; accessed 04-May-2010].

[11] Spinal cord injury. http://www.spinalcord.org/. [Online; accessed 13-April-2010].

[12] Upnp forum. http://www.upnp.org/. [Online; accessed 20-May-2010].

[13] World Health Organization. http://www.who.int/mental_health/neurology/neurodiso/en/index.html. [Online; accessed 04-April-2010].

[14] Yuska Paola Costa Aguiar. *Protocolo experimental para análise da interação entre usuários e sistemas, com foco na automação industrial*. Exame de Qualificação, Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, Maio 2010.

[15] S. Barrera, H. Takahashi, and M. Nakajima. Hands-free navigation methods for moving through a virtual landscape walking interface virtual reality input devices. In *Computer Graphics International, 2004. Proceedings*, pages 388 –394, june 2004.

[16] N. Birbaumer, T. Hinterberger, A. Kubler, and N. Neumann. The thought-translation device (TTD): neurobehavioral mechanisms and clinical outcome. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 11(2):120 –123, june 2003.

[17] N. Birbaumer, A. Kubler, N. Ghanayim, T. Hinterberger, J. Perelmouter, J. Kaiser, I. Iversen, B. Kotchoubey, N. Neumann, and H. Flor. The thought translation device (TTD) for completely paralyzed patients. *Rehabilitation Engineering, IEEE Transactions on*, 8(2):190 –193, jun 2000.

[18] Andreas Bulling, Daniel Roggen, and Gerhard Tröster. It's in your eyes: towards context-awareness and mobile HCI using wearable EOG goggles. In *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*, pages 84–93, New York, NY, USA, 2008. ACM.

[19] Andreas Bulling, Daniel Roggen, and Gerhard Tröster. Wearable EOG goggles: eye-based interaction in everyday environments. In *CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 3259–3264, New York, NY, USA, 2009. ACM.

[20] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[21] Gong Chao. Human-computer interaction: The usability test methods and design principles in the human-computer interface design. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, pages 283 –285, 8-11 2009.

[22] Rory A. Cooper and Rosemarie Cooper. *Chalanges for Assistive Technologies*, chapter The Potential of Technology to Improve Quality of Life, pages 8 – 14. IOS Press, 2007.

[23] M. Cope, P. van der Zee, S.R. Arridge, M. Essenpreis, C.E. Elwell, and D.T. Delpy. Non-invasive measurement of tissue oxygenation using near infrared (NIR) spectroscopy. In

Optical Techniques and Biomedical Applications, IEE Colloquium on, pages 14/1 –14/4, jun 1991.

[24] N.S. Dias, R.M. Mendes, and J.H. Correia. Subject age in P300 BCI. In *Neural Engineering, 2005. Conference Proceedings. 2nd International IEEE EMBS Conference on*, pages 579 –582, march 2005.

[25] T. Ebrahimi, J.-M. Vesin, and G. Garcia. Brain-computer interface in multimedia communication. *Signal Processing Magazine, IEEE*, 20(1):14 – 24, jan 2003.

[26] G. Edlinger, C. Holzner, C. Guger, C. Groenegress, and M. Slater. Brain-computer interfaces for goal orientated control of a virtual smart home environment. pages 463 –465, 29 2009-may 2 2009.

[27] L. A. Farwell and E. Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 70(6):510–523, December 1988.

[28] T. Hinterberger, J. Mellinger, and N. Birbaumer. The thought translation device: structure of a multimodal brain-computer communication system. pages 603 – 606, march 2003.

[29] T. Hinterberger, S. Schmidt, N. Neumann, J. Mellinger, B. Blankertz, G. Curio, and N. Birbaumer. Brain-computer communication and slow cortical potentials. *Biomedical Engineering, IEEE Transactions on*, 51(6):1011 –1018, june 2004.

[30] A.A. Ioannides and J.G. Taylor. Testing models of attention with MEG. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 1, pages 287 – 291 vol.1, july 2003.

[31] ISO, editor. *International Organization or Standardization, Ergonomic equirements for office work with visual display terminals*. 1998.

[32] V. Kecman. *Support Vector Machines:Theory and Applications*, chapter Support Vector Machines An Introduction, pages 1 – 47. Springer-Verlag Berlin Heidelberg, 2005.

[33] Roman Krepki. *Design and implementation of an online BCI system for the control in gaming applications and virtual limbs.* PhD dissertation, Berlin University, Department of Electrical engineering and computer science, Nov 2004.

[34] Roman Krepki, Benjamin Blankertz, Gabriel Curio, and Klaus-Robert Müller. The berlin brain-computer interface (BBCI) — towards a new communication channel for online control in gaming applications. *Multimedia Tools Appl.*, 33(1):73–90, 2007.

[35] Roman Krepki, Gabriel Curio, Benjamin Blankertz, and Klaus-Robert Müller. Berlin brain-computer interface-the HCI communication channel for discovery. *Int. J. Hum.-Comput. Stud.*, 65(5):460–477, 2007.

[36] Andrea Kubler and Klaus-Robert Muller. *Toward Brain-Computer Interfacing*, chapter An Introduction to Brain-Computer Interfacing, pages 1 – 25. The MIT Press, 2007.

[37] Chia-Wei Lee, Der-Yow Chen, Chang-Wei Wu, and Jyh-Horng Chen. Comparing the spatial and temporal reproducibility of brain activation using three fMRI techniques: BOLD, FAIR, and VASO. In *Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging, 2007. NFSI-ICFBI 2007. Joint Meeting of the 6th International Symposium on*, pages 258 –261, oct. 2007.

[38] Hyeopwoo Lee, Sukmoon Chang, Dongsuk Yook, and Yongserk Kim. A voice trigger system using keyword and speaker recognition for mobile devices. *Consumer Electronics, IEEE Transactions on*, 55(4):2377 –2384, november 2009.

[39] Dennis J. McFarland, Dean J. Krusienski, and Jonathan R. Wolpaw. Brain-computer interface signal processing at the wadsworth center: mu and sensorimotor beta rhythms. In Christa Neuper and Wolfgang Klimesch, editors, *Event-Related Dynamics of Brain Oscillations*, volume 159 of *Progress in Brain Research*, pages 411 – 419. Elsevier, 2006.

[40] G.R. Muller-Putz and G. Pfurtscheller. Control of an electrical prosthesis with an SSVEP-based BCI. *Biomedical Engineering, IEEE Transactions on*, 55(1):361 –364, jan. 2008.

[41] Anton Nijholt and Desney Tan. Brain-computer interfacing for intelligent systems. *IEEE Intelligent Systems*, 23(3):72–79, 2008.

[42] Martin Oehler, Peter Neumann, Matthias Becker, Gabriel Curio, and Meinhard Schilling. Extraction of SSVEP signals of a capacitive EEG helmet for human machine interface. pages 4495 –4498, aug. 2008.

[43] M.A. Oskoei and Huosheng Hu. Application of feature tracking in a vision based human machine interface for Xbox. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pages 1738 –1743, dec. 2009.

[44] José C. Principe and Dennis J. McFarland. *Brain-Computer Interfaces An International Assesment of Research an Development Trends*, chapter BMI/BCI Modeling and Signal Processing, pages 47 – 64. Springer, 2008.

[45] Genaro Rebolledo-Mendez, Ian Dunwell, Erika A. Martínez-Mirón, María Dolores Vargas-Cerdán, Sara Freitas, Fotis Liarokapis, and Alma R. García-Gaona. Assessing

neurosky's usability to detect attention levels in an assessment exercise. In *Proceedings of the 13th International Conference on Human-Computer Interaction. Part I*, pages 149–158, Berlin, Heidelberg, 2009. Springer-Verlag.

[46] T. Sales, L. Sales, M. Pereira, H. Almeida, A. Perkusich, K. Gorgonio, and M.A. de Sales. Towards the upnp-up: Enabling user profile to support customized services in upnp networks. In *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM '08. The Second International Conference on*, pages 206 –211, sept. 2008.

[47] N. Sambasivan and M. M. Jackson. Designing pervasive brain-computer interfaces. pages 267–272, 2007.

[48] A. Schlogl and C. Brunner. Biosig: A free and open source software library for BCI research. *Computer*, 41(10):44–50, 2008.

[49] Mark Summerfield. *Rapid GUI Programming with Python and Qt*, chapter Appendix A. Installing, pages 561 – 574. Prentice Hall, 2007.

[50] Angelo Perkusich Hyggo Almeida Maurus Holanda Taciana Rached, Danilo Santos. Bci-aware pervasive multimedia for motor disabled people. *International Conference on Information Society, IEEE*, june 2010.

[51] Angelo Perkusich Hyggo Almeida Maurus Holanda Taciana Rached, Danilo Santos. Controlling interactive digital television applications by brain-computer interface. *8th European Conference on Interactive TV and Video, ACM*, june 2010.

[52] E. Teo, A. Huang, Y. Lian, C. Guan, Y. Li, and H. Zhang. Media communication center using brain computer interface. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 2954–2957, 30 2006-Sept. 3 2006.

[53] J. Wang, N. Yan, H. Liu, and C. Tai. Brain-computer interfaces based on attention and complex mental tasks. *Clinical Neurophysiology*, 1(4), March 2008.

[54] Yijun Wang, Ruiping Wang, Xiaorong Gao, Bo Hong, and Shangkai Gao. A practical VEP-based brain-computer interface. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 14(2):234 –240, june 2006.

[55] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):66–75, September 1991.

[56] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 1:767–791, March 2002.

[57] Ed Zintel. Tools and products. *Computer Graphics and Applications, IEEE*, 28(4):103 –104, july-aug. 2008.

Apêndice A

Artefatos Gerados Para a Realização de Teste de Usabilidade

Neste Apêndice, são apresentados os artefatos gerados a partir do protocolo experimental discutido na Seção 3.9. Esses artefatos incluem o plano geral do ensaio de avaliação, o plano de avaliação com usuário, o plano de coleta de dados, o roteiro de tarefa de testes, a ficha de observação, questionário pré-teste, questionário pós-teste, ficha de cadastro e o termo de aceitação das condições de realização do teste.