



UFRR

UNIVERSIDADE FEDERAL DE RORAIMA  
PRÓ-REITORIA DE ENSINO E GRADUAÇÃO  
CENTRO DE CIÊNCIA E TECNOLOGIA  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

JOÃO PAULO VERÇOSA PINTO

HAND.IO: UMA LUVA PARA CONTROLE DE DISPOSITIVOS ELETRO-ELETRÔNICOS  
UTILIZANDO RECONHECIMENTO DE GESTOS

Boa Vista - RR

2018

JOÃO PAULO VERÇOSA PINTO

HAND.IO: UMA LUVA PARA CONTROLE DE DISPOSITIVOS ELETRO-ELETRÔNICOS  
UTILIZANDO RECONHECIMENTO DE GESTOS

Trabalho de conclusão de curso na área  
de Controle e Automação desenvolvido  
na UFRR com o objetivo de facilitar o  
controle de dispositivos através de gestos

Orientador: Dr. Herbert Oliveira Rocha

Boa Vista - RR

2018

JOÃO PAULO VERÇOSA PINTO

**HAND.IO: UMA LUVA PARA CONTROLE DE DISPOSITIVOS  
ELETRO-ELETRÔNICOS UTILIZANDO RECONHECIMENTO DE  
GESTOS**

Trabalho de conclusão de curso na área  
de Controle e Automação desenvolvido na  
UFRR com o objetivo de facilitar o controle  
de dispositivos através de gestos

Aprovado pela banca examinadora em 12 de Julho de 2018:

---

**Prof. DSc. Herbert Oliveira Rocha**  
Orientador

---

**Prof. DSc. Leandro Nelinho Balico**  
Membro da Banca

---

**Prof. MSc. Acauan Cardoso Ribeiro**  
Membro da Banca

## *Dedicatória*

# AGRADECIMENTOS

*A todos aqueles tops*

# RESUMO

**Palavras-chaves:** controle de dispositivos, acelerômetro, giroscópio, ambientes inteligentes, vestíveis, luva inteligente.

# LISTA DE FIGURAS

Figure 1 – Diagrama de um sistema em tempo real genérico. (Adaptado de Buttazzo (2011)) . . . . .	16
Figure 2 – Visão geral do protótipo . . . . .	27
Figure 3 – Fluxograma da Hand.io . . . . .	28
Figure 4 – Esquemático do protótipo da luva . . . . .	31
Figure 5 – Esquemático da central de processamento de sinais e execução de ações . . . . .	31



# LISTA DE TABELAS

Table 1 – Comparação entre Hand.io e SoapBox . . . . .	24
Table 2 – Comparação entre Hand.io e uWave . . . . .	26
Table 3 – Objetivo do estudo. . . . .	34

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>1.1</b>	<b>Definição do Problema</b>	<b>12</b>
<b>1.2</b>	<b>Objetivos</b>	<b>12</b>
1.2.1	Objetivos Específicos	12
<b>2</b>	<b>CONCEITOS E DEFINIÇÕES</b>	<b>14</b>
<b>2.1</b>	<b>Sistemas Embarcados</b>	<b>14</b>
2.1.1	Restrições de um Sistema Embarcado	14
2.1.2	Sistemas de Tempo Real	16
2.1.3	Microcontroladores e Microprocessadores	17
<b>2.2</b>	<b>Modelagem e verificação de sistemas</b>	<b>18</b>
2.2.1	United Modeling Language	19
2.2.2	Modelos Formais	19
2.2.3	Máquina de Estados	19
2.2.4	Rede de Petri	19
<b>2.3</b>	<b>Internet das Coisas</b>	<b>19</b>
<b>2.4</b>	<b>Reconhecimento de Padrões</b>	<b>20</b>
2.4.1	Aprendizagem de Máquina	20
<b>3</b>	<b>TRABALHOS CORRELATOS</b>	<b>22</b>
<b>3.1</b>	<b>Accelerometer-based gesture control for a design environment</b>	<b>23</b>
<b>3.2</b>	<b>I'm home: Defining and evaluating a gesture set for smart-home control</b>	<b>24</b>
<b>3.3</b>	<b>uWave: Accelerometer-based personalized gesture recognition and its applications</b>	<b>25</b>
<b>4</b>	<b>MÉTODO PROPOSTO</b>	<b>27</b>
<b>4.1</b>	<b>Visão geral do método da Hand.io</b>	<b>27</b>
<b>4.2</b>	<b>Fluxo de execução da Hand.io</b>	<b>28</b>
<b>4.3</b>	<b>Captura de Dados Baseado em Movimentos de Amplitude de Punho e Mão</b>	<b>29</b>
4.3.1	Conexão com a central de processamento	29
<b>4.4</b>	<b>Reconhecimento de Padrões Baseado em Movimentos e Ações</b>	<b>29</b>
<b>4.5</b>	<b>Modelo de Conexão Entre Dispositivos Eletrônicos</b>	<b>30</b>
<b>4.6</b>	<b>Modelo de Prototipação</b>	<b>30</b>

5	RESULTADOS PARCIAIS . . . . .	32
6	CONSIDERAÇÕES PARCIAIS E PRÓXIMO PASSOS . . . . .	33
7	APÊNDICE 1: REVISÃO SISTEMÁTICA DA LITERATURA . . . . .	34
7.1	Planejamento da Revisão Sistemática da Literatura . . . . .	34
7.1.1	Procedimentos de Seleção e Critérios . . . . .	36
	BIBLIOGRAPHY . . . . .	38

# 1 INTRODUÇÃO

A complexidade de sistemas computacionais cresce exponencialmente, visando aplicações em diferentes domínios, tais como: agricultura, automação residencial e robótica. Neste sentido, Sistemas Embarcados (SE) que são sistemas computacionais integrados a outros sistemas, usualmente são encontrados no nosso dia a dia. Geralmente, o principal propósito dos SE é o controle e provimento de informações para uma função específica (RAMESH et al., 2012) sendo estes extremamente iterativos com seu ambiente, operam geralmente em tempo real, e estão disponíveis continuamente.

Segundo Yurish (2016), robôs se tornam mais inteligentes com respostas em tempo real. Sistemas como o Controle Integrado de Forças processam as variações com sensibilidade humana, melhorando a performance, e reduzindo assim o tempo de codificação. Isso torna os robôs mais inteligentes e capazes, como os humanos, de manusear itens com interações externas em tempo real. Robôs podem auxiliar pessoas com restrições motoras a realizar atividades cotidianas que muitas vezes envolvem o contato com o rosto de uma pessoa.

Levando em consideração estes avanços no campo de sensores robóticos, desenvolver sistemas embarcados com características robóticas, por exemplo, a capacidade de interação com o ambiente, pode ser uma solução viável para que humanos possam interagir com dispositivos eletro-eletrônicos (splints de ar-condicionado, televisões, computadores e outros) ao seu redor sem a necessidade do toque presencial. Um modo de tornar esta solução viável seria a utilização de luvas inteligentes, as quais contam com utilização de: acelerômetros, sensores de força, um microcontrolador e transmissores (NAVAS et al., 2012) para reconhecimento de padrões e então comunicação com os dispositivos eletro-eletrônicos (O'FLYNN et al., 2013) (BERNIERI et al., 2015) (CHOUDHARY et al., 2015). Adicionalmente, tais características robóticas podem auxiliar na recuperação de doenças ou mesmo prover suporte complementar em casos de necessidades especiais dos humanos, exemplo, deficiência visual.

No trabalho de Choudhary et al. (2015) é proposta uma nova abordagem para auxiliar a comunicação e a interação de indivíduos surdos-mudos, aumentando sua independência. Este inclui uma luva inteligente que traduz o alfabeto Braille, que é o mais universalmente utilizado pela população surdo-muda alfabetizada, em texto e vice-versa, e comunica mensagens via SMS para contato remoto. Isto permite ao usuário transmitir mensagens simples através de sensores de toque capacitivos como sensores de entrada colocados na palma da luva e convertidos para texto pelo computador/telefone móvel. O usuário pode entender e interpretar mensagens recebidas através de padrões de retornos

táteis de pequenos motores de vibração no dorso da luva. A implementação bem-sucedida da tradução bidirecional em tempo real entre Inglês e Braille, e a comunicação entre o dispositivo vestível e o computador/telefone móvel abrem novas oportunidades para a troca de informação que até então estavam indisponíveis para indivíduos surdos-mudos, como a comunicação remota, assim como transmissões paralelas de um pra muitos.

O'Flynn et al. (2013) apresenta o desenvolvimento de uma luva inteligente para facilitar o processo de reabilitação de Artrite Reumatoide através da integração de sensores, processadores e tecnologia sem fio para medir empiricamente o alcance do movimento. A Artrite Reumatoide é uma doença que ataca o tecido sinovial lubrificando as juntas do esqueleto. Esta condição sistêmica afeta os sistemas esquelético e muscular, incluindo ossos, juntas, músculos e tendões o que contribui para a perda de função de alcance do movimento. Medições tradicionais de artrite precisam de exames pessoais intensamente trabalhosos realizados por uma equipe médica que através de suas medições objetivas podem dificultar a determinação e a análise da reabilitação da artrite. A luva proposta (O'FLYNN et al., 2013) usa uma combinação de 20 sensores de dobras, 16 acelerômetros de três eixos, e 11 sensores de força para detectar o movimento de juntas. Todos os sensores são posicionados em um PCB flexível para permitir um alto nível de flexibilidade e estabilidade do sensor.

Analisando o processo de desenvolvimento destes sistemas, verifica-se que existem diferentes graus de complexidades tanto no software quanto no hardware. Desta forma, é necessário que as aplicações sejam projetadas considerando os requisitos de previsibilidade e confiabilidade, principalmente em aplicações de sistemas embarcados críticos, onde diversas restrições (por exemplo, tempo de resposta e precisão dos dados) devem ser atendidas e mensuradas de acordo com os requisitos do usuário, caso contrário uma falha pode conduzir a situações catastróficas. Por exemplo, o erro de cálculo da dose de radiação no Instituto Nacional de Oncologia do Panamá que resultou na morte de 23 pacientes (WONG et al., 2010). Contudo, erros durante o desenvolvimento de sistemas computacionais tornam-se mais comuns, principalmente quando se tem curto espaço de tempo de liberação do produto ao mercado, e estes sistemas precisam ser desenvolvidos rapidamente e atingir um alto nível de qualidade.

Visando contribuir com metodologias de desenvolvimento de sistemas embarcados, o contexto deste trabalho está situado em demonstrar métodos e técnicas no processo de codificação e prototipação de sistemas embarcados, tais como: Redes de Petri (BENDERS; STEVENS, 1992) (GIRAULT; VALK, 2002), UML (CUNHA et al., 2011); e Máquinas de Estados (LAMPKA et al., 2009).

O desenvolvimento de tal sistema integrado à vestimenta do usuário funcionará de forma ubíqua e pervasiva, o que permitirá uma melhor interação com ambientes inteligentes, que em um curto período de tempo se tornarão padrão nas áreas residenciais e ambientes de convívio público. Segundo Weiser (1999) as redes ubíquas já são capazes de realizar

Adicionar  
refer-  
ência

Adicionar  
de-  
scrição  
da  
sigla

Falta  
conec-  
tar  
este  
texto  
ao an-  
terior

certas operações em um contexto onde diversos dispositivos inteligentes estão em constante comunicação.

Neste contexto, este trabalho visa o desenvolvimento de uma luva inteligente de baixo custo para controle de dispositivos eletrônicos por meio do reconhecimento de padrões de movimentos, visto que tais aplicações se mostra de grande utilidade para o usuário em um ambiente inteligente, assim provendo conforto e facilidades ao seu usuário. Adicionalmente, para o desenvolvimento do sistema proposto, este trabalho visa projetar e analisar componentes de hardware e software com o foco em IoT que tenham um baixo consumo de energia; baixo custo de implementação; e aplicação de métodos para garantir a qualidade da execução do sistema computacional proposto.

## 1.1 Definição do Problema

O problema considerado neste trabalho é expresso na seguinte questão: Como projetar e desenvolver um sistema embarcado, para uma luva inteligente de baixo custo, utilizando métodos e técnicas para garantir os requisitos de previsibilidade e confiabilidade do sistema, de tal forma que o sistema proposto auxilie o seu usuário na comunicação com dispositivos eletrônicos?

## 1.2 Objetivos

O objetivo principal deste trabalho é demonstrar métodos e técnicas no processo de codificação e prototipação de sistemas embarcados em uma luva inteligente, visando garantir os requisitos de previsibilidade; confiabilidade; e baixo custo para a comunicação com dispositivos eletrônicos por meio do reconhecimento de padrões de gestos. Assim, visando a criação de uma interface única e intuitiva entre um usuário e um ambiente inteligente, visando facilitar o uso de dispositivos através de gestos, que são um meio natural de comunicação.

### 1.2.1 Objetivos Específicos

1. Identificar métodos para a modelagem do software e hardware;
2. Definir um modelo formal para o fluxo do funcionamento da luva, visando analisar propriedades de segurança do fluxo de execução do sistema;
3. Demonstrar uma técnica para transformação de modelos de software em códigos para o projeto;

adicionar  
sobre  
cidades  
in-  
teligentes  
inter-  
net  
das  
coisas  
com-  
putação  
perva-  
siva e  
ubíqua

4. Projetar e desenvolver uma central de controle que será o meio de comunicação entre os dispositivos eletrotônicos disponíveis em um ambiente com e a luva para obtenção dos dados de movimentos;
5. Identificar e aplicar um protocolo de comunicação entre os dispositivos eletrotônicos disponíveis em um ambiente com a luva;
6. Realizar o levantamento de componentes eletrônicos com baixo consumo de energia e baixo custo necessários para o desenvolvimento de um protótipo;
7. Propor um algoritmo para reconhecimento e classificação dos movimentos/sinais enviados pela luva na mão do usuário;
8. Desenvolver um protótipo da luva e da central de controle; e
9. Validar o método proposto pela análise da prototipação do sistema proposto, a fim de examinar a sua eficácia e aplicabilidade.

## 2 CONCEITOS E DEFINIÇÕES

Este capítulo tem como objetivo apresentar as principais conceitos e definições que serão utilizados para o entedimento e desenvolvimento deste trabalho.

### 2.1 Sistemas Embarcados

De acordo com [Vahid e Givargis \(2001\)](#) um sistema embarcado é um sistema computacional desenvolvido para um propósito específico, que em contraposição a sistemas de propósito geral, realiza apenas uma tarefa específica. Existem certas características (exemplo, restrição de memória e tempo de execução) que distinguem um sistema embarcado dos demais, que apesar de nem sempre serem atendidas, servem de referência para a classificação de tais sistemas. Sistemas embarcados geralmente realizam apenas uma atividade repetidas vezes, contam com restrições mais apertadas que um sistema normal, são reativos e muitas vezes funcionam em tempo real. Estes sistemas geralmente fazem parte de sistemas maiores, e quase sempre funcionam sem o conhecimento do usuário.

Dispositivos como *smartphones* e computadores pessoais, são compostos por diversos sistemas embarcados que realizam apenas uma função, mas que quando estão juntos passam a realizar diversas atividades, passando a ser classificado como um sistema de propósito geral. Um exemplo de um sistema embarcado que faz parte de um sistema maior seria a placa de rede de um smartphone ([VIVE... , 2017](#)), que por si só é um sistema que apenas recebe e envia dados através de uma rede sem fio, mas que quando contextualizado em um smartphone possibilita o acesso a internet, o que promove a realização de diversas atividades disponíveis online, como acesso à redes sociais ou a reprodução de vídeos ([VAHID; GIVARGIS, 2001](#)).

Corrigir  
citação  
com  
...

Falar sobre IoT

#### 2.1.1 Restrições de um Sistema Embarcado

Um sistema embarcado geralmente trabalha em condições restritas nas quais certas métricas devem ser cumpridas para que seu funcionamento se dê de maneira eficiente. [Marwedel \(2011\)](#) define uma série de métricas utilizadas para medir a eficiência de um dado sistema como: consumo energético, eficiência em tempo de execução, tamanho de código, peso e custo. Em alguns casos existem situações onde estas métricas entram em conflito, neste caso cabe ao projetista do sistema definir qual a melhor configuração abrindo mão de certas funcionalidades para melhor atender as necessidades da aplicação.



O consumo de energia em um sistema embarcado muitas vezes é limitado devido à tecnologia atual de baterias e ao tipo do circuito utilizado. Circuitos Integrados de Aplicações Específicas (ASICs, em inglês) tendem a ter uma eficiência energética muito superior aos demais, no entanto abrem mão de flexibilidade no desenvolvimento do software, circuitos reconfiguráveis como os Arranjos de Portas Programáveis em Campo (FPGAs, em inglês) tem sua eficiência energética inferior aos ASICs, mas oferecem uma flexibilidade de desenvolvimento muito maior apesar de serem limitados pelo tamanho dos circuitos reconfiguráveis disponíveis (MARWEDEL, 2011).

A limitação entre flexibilidade e eficiência também se aplica aos processadores. Processadores desenvolvidos especificamente para processamento de sinais analógicos, por exemplo, são exponencialmente mais eficientes energeticamente que processadores de propósito geral, que contam com a pior eficiência energética dentre os circuitos apresentados, mas permitem uma grande gama de possibilidades durante o desenvolvimento do sistema (MARWEDEL, 2011).

As limitações em tempo de execução devem ser reduzidas ao máximo para aproveitar o hardware disponível da melhor maneira possível. Problemas acarretados por compiladores que geram binários que não utilizam todo o potencial da arquitetura devem ser corrigidos a fim de eliminar desperdícios de instruções por ciclo de processamento (MARWEDEL, 2011).

O tamanho do código e as limitações em espaço de armazenamento são um desafio recorrente no mundo dos sistemas embarcados. Em diversas aplicações onde não existe a possibilidade de carregamento dinâmico de dados, como ocorrem nos *smartphones*, todos os dados devem ser armazenados na memória limitada dos chips, como nos casos de *Systems on a Chip* (SoC), nos quais todos os componentes de processamento de dados se encontram em um único chip. Nesse tipo de situação onde o espaço se torna um recurso precioso é necessário um cuidado maior durante a construção do código que será executado na plataforma. Tal limitação pode ser contornada com a utilização de memórias *flash*, mas em certas aplicações devido à outras limitações este tipo de recurso pode não estar disponível (MARWEDEL, 2011).

O peso dos componentes pode ser o fator decisivo na aplicação de um sistema embarcado, existem situações onde o sistema deve ser portátil, logo um peso elevado acabaria por dificultar o manuseio do protótipo. Nestes casos o projetista deve ter preferência pelos componentes de tamanho mais reduzido possível (MARWEDEL, 2011).

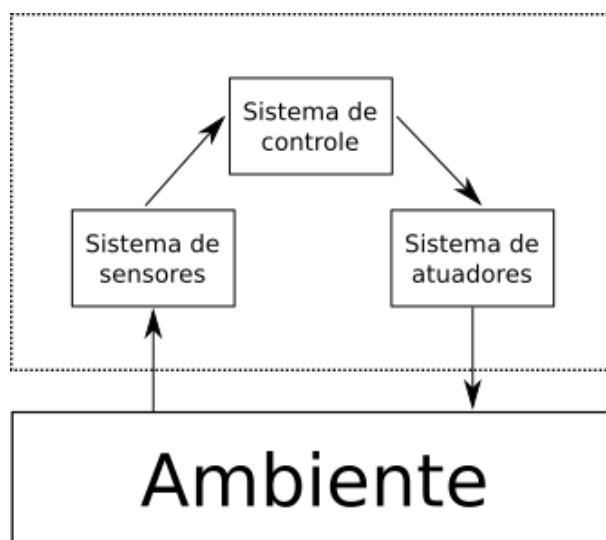
Em sistemas voltados para o mercado de consumo onde existe um orçamento limitado para o desenvolvimento do sistema, o custo passa a ser uma métrica crucial para o projeto, componentes que não melhoram de maneira significativa a eficiência do pior caso do sistema devem ser descartados afim de reduzir o consumo total de energia. Os requisitos da aplicação devem ser cumpridos utilizando o menor número de componentes

possível e com os componentes mais baratos possíveis desde que não comprometam a qualidade do produto final. (MARWEDEL, 2011)

Descrever que algumas destas restrições são críticas e precisam de validação com método formais

## 2.1.2 Sistemas de Tempo Real

Conforme Buttazzo (2011), um sistema em tempo real deve ter o tempo do sistema medido utilizando a mesma escala de tempo do mundo real, isto ocorre pelo fato de que o sistema deve ter ciência do ambiente no qual ele irá operar. Buttazzo (2011) realiza uma comparação entre sistemas biológicos e a velocidade das suas reações em seus habitats, um gato e uma tartaruga, por exemplo, podem ter velocidades de reação diferentes, no entanto em seus respectivos ambientes esta velocidade se mostra suficiente para a sua sobrevivência. Tal exemplo demonstra que o conceito de tempo não é natural aos sistemas, sejam eles biológicos ou artificiais, mas que na verdade está relacionado com o ambiente no qual estes sistemas irão atuar como apresentado na Figura 1.



Explicar a figura

Figure 1 – Diagrama de um sistema em tempo real genérico. (Adaptado de Buttazzo (2011))

Em termos de processamento, a computação mais rápida reduz o tempo de resposta de um sistema, entretanto ela não necessariamente garante que o tempo de resposta de tarefas individuais será atingido de maneira correta, um sistema em tempo real não deve apenas ser rápido, ele deve ser previsível. Analisando os sistemas do ponto de vista de processos, os processos de um sistema em tempo real contam com um componente ausente em processos de sistemas normais, o chamado *Deadline*, que é o prazo máximo para a finalização de uma determinada tarefa. Em aplicações críticas o retorno de operações fora do *deadline* não é apenas atrasado, mas sim incorreto, o que pode ocasionar em perdas significativas dependendo da criticidade do sistema (BUTTAZZO, 2011).

Exemplificar perda de dead-

A criticidade de uma aplicação depende das consequências ocasionadas devido ao atraso no tempo de resposta esperado do sistema, sendo este classificado em *Hard*, *Firm* e *Soft*.

- Sistemas críticos *Hard* são aqueles onde caso não haja resposta no tempo definido podem ocorrer eventos devastadores, muitas vezes com perda de vidas;
- Sistemas críticos *Firm* são aqueles onde o atraso na resposta torna o sistema inútil, no entanto nenhum dano é gerado; e
- Sistemas críticos *Soft* são aqueles onde resultados após o tempo de resposta definido ainda tem alguma utilidade para o sistema, mesmo gerando perda de desempenho.

Usualmente os sistemas embarcados trabalham de maneira híbrida quanto a sua criticidade, onde certas atividades podem ser consideradas como *Hard* e outras podem ser *Soft* ou *Firm*. Atividades com criticidade *Hard* incluem: coleta de dados utilizando sensores, detecção de condições críticas, filtragem de dados, etc. Atividades que contam com uma criticidade *Firm* podem ser encontradas em aplicações de redes e multimídia, por exemplo: processamento de imagem *on-line*, execução de vídeos e decodificação de áudio e vídeo. Já atividades com criticidade *Soft* geralmente estão relacionadas à interação com o usuário como: a exibição de mensagens em uma tela, o processamento de sinais de teclado e o armazenamento de dados de utilização. (BUTTAZZO, 2011)

### 2.1.3 Microcontroladores e Microprocessadores

As diferenças entre microcontroladores e microprocessadores são apresentadas por Ayala (1991) em seu livro, onde fica claro que apesar de terem surgido da mesma ideia e serem fabricados pelo mesmo grupo de pessoas, seu funcionamento e aplicação diferem grandemente. Existem diferenças fundamentais no design de tais dispositivos, enquanto microcontroladores por si só contam com todos os componentes necessários para o seu funcionamento, microprocessadores necessitam de outros componentes (exemplo, memória) e periféricos para funcionar como um computador completo.

Microprocessadores são conhecidos popularmente como Unidades Centrais de Processamento (CPU, em inglês), e tem como sua função principal buscar e modificar extensivamente dados da memória para que sejam armazenados ou exibidos para o usuário. Por si só um microprocessador não constitui um microcomputador completo, para se tornar uma máquina capaz de executar programas de propósito geral se faz necessária a utilização de memórias RAM, memórias de armazenamento massivo e diversos dispositivos de entrada e saída externos (AYALA, 1991).

Nos microprocessadores o hardware destes componentes é desenvolvido de tal maneira a permitir o desenvolvimento de sistemas grandes ou pequenos dependendo da

demanda da aplicação. O projeto do microprocessador é desenvolvido de modo a atender as suas expectativas no mercado de consumo em massa (AYALA, 1991).

Os microcontroladores funcionam de maneira similar aos microprocessadores, no entanto, além de terem os componentes encontrados em um microprocessador eles também contam com todos os componentes necessários para o funcionamento de um computador completo, como memórias RAM, memórias ROM e portas paralelas e seriais de entrada e saída. Como os microprocessadores, os microcontroladores também são componentes de propósito geral, mas o seu foco deixa de ser apenas computar os dados encontrados na memória e armazená-los, e passa também controlar o ambiente em que se encontram se baseando nos resultados do processamento dos dados disponíveis (AYALA, 1991).

Os programas utilizados em microcontroladores são armazenados na memória ROM e não tem seu funcionamento alterado durante o ciclo de vida do sistema. As instruções de máquina encontradas nos microcontroladores geralmente envolvem buscar dados na memória interna e que também realizam operações envolvendo os pinos de conexão inclusos na plataforma, o que permite que cada pino tenha seu propósito programado de acordo com a vontade do desenvolvedor (AYALA, 1991).

## 2.2 Modelagem e verificação de sistemas

Segundo Edwards et al. (1997), os sistemas atualmente, em sua maioria, são desenvolvidos com uma abordagem *ad hoc* que é baseada em experiências prévias dos desenvolvedores com produtos similares aos que serão utilizados durante a implementação do sistema. Uma abordagem utilizando modelos formais e métodos de síntese automatizada de implementação é uma maneira mais efetivas de garantir um sistema seguro, visto que tais abordagens adotam modelos matemáticos para certificar propriedades de execução do sistema desenvolvido.

Um modelo é uma representação abstrata e simplificada de uma coisa real. No desenvolvimento de sistemas um modelo permite aos desenvolvedores a possibilidade de distinguir o que realmente é necessário e o que é supérfluo, além de permitir que seja realizada uma análise mais rápida do sistema do que se cada componente do sistema tivesse que ser analisado individualmente (MILES, 2006).

A utilização de um ou mais modelos formais para descrever os comportamentos do sistema antes de tomar decisões mais específicas da implementação, como por exemplo, qual plataforma de hardware ou qual linguagem de programação utilizar, podem reduzir consideravelmente as falhas durante o funcionamento do sistema (EDWARDS et al., 1997).

Estes modelos permitem que as propriedades do sistema sejam validadas de maneira simples, ainda durante as fases iniciais do desenvolvimento, através de ferramentas de

Apresenta  
um  
exem-  
plo de  
micro-  
pro-  
ces-  
sador

Apresenta  
um  
exem-  
plo de  
micro-  
con-  
tro-  
lador  
com  
a de-  
scrição  
da  
pinagem

simulação e verificação automatizadas, evitando eventuais entraves que podem comprometer a eficiência do sistema (EDWARDS et al., 1997).

### 2.2.1 United Modeling Language

De acordo com Miles (2006) a Linguagem de Modelagem Unificada (UML, em Inglês) é um padrão amplamente utilizado por desenvolvedores para modelagem de software e hardware de sistemas em geral. A UML é uma ferramenta muito útil durante o desenvolvimento de um sistema, pois permite aos desenvolvedores manter um certo controle sobre a complexidade do sistema, permitindo que as partes mais importantes do sistema se sobressaiam em relação às outras, evitando eventuais confusões durante a implementação.

Apresentar os diagramas com exemplo

Descrever um exemplo com diagrama de sequência e apresentar como será usado no seu TCC

### 2.2.2 Máquina de Estados

Segundo Hopcroft et al. (2001) máquinas de estados finitos, também conhecidas como autômatos finitos, são uma maneira muito útil de modelar um sistema de maneira que seus componentes ou subsistemas estejam sempre realizando um conjunto de tarefas finitas que podem ser representadas e rastreadas através de estados que são alterados em resposta à entradas externas.

O propósito do estados é que o sistema tenha noção do seu histórico, se lembrando apenas das partes relevantes.

### 2.2.3 Rede de Petri

Girault e Valk (2002)

## 2.3 Internet das Coisas

No trabalho de Atzori et al. (2010) é demonstrado que em sua ideia fundamental a internet das coisas é um ambiente de computadores pervasivos que interagem um com os outros de maneira cooperativa para atingir metas em comum, este conceito permite a idealização de ambientes inteligentes onde todos os objetos estão em constante comunicação: para auxiliar o usuário a atingir uma qualidade de vida maior.

Nos ambientes residenciais e empresariais, a internet das coisas pode tornar a vida muito mais confortável e eficiente com sensores e atuadores, coletando dados sobre

a preferência dos usuários e sobre o clima é possível que a temperatura seja adequada automaticamente levando em consideração o gasto de energia elétrica para uma maior economia.

## 2.4 Reconhecimento de Padrões

Segundo Bishop (2006) o problema de encontrar padrões em dados é um problema famoso e antigo que desde sempre intrigou cientistas mundo a fora. A observação de padrões na natureza serviu de fundação para a criação de modelos matemáticos que servem de base para a física moderna que até hoje é utilizada para resolver problemas reais.

Atualmente o campo de reconhecimento de padrões é focado na identificação automática de regularidades em grandes conjuntos de dados, utilizando algoritmos computadorizados. Os resultados destes algoritmos podem ser utilizados para classificar estas regularidades em diferentes grupos. (BISHOP, 2006)

### 2.4.1 Aprendizagem de Máquina

Segundo Russell e Norvig (2010) o aprendizado ocorre quando um agente, seja ele uma máquina ou um humano, melhora seu desempenho em tarefas futuras após realizar observações do ambiente no qual ele atua se baseando em pares de entradas e saídas para prever novas saídas a partir de novas entradas.

A utilização do aprendizado de máquina se justifica pela necessidade de criar sistemas adaptativos, pois durante o desenvolvimento muitas vezes não é possível prever mudanças que podem ocorrer no ambiente no qual o sistema irá atuar, ou mesmo porque não é possível de maneira simples desenvolver um sistema que possa atender as necessidades da aplicação utilizando métodos tradicionais de programação. (RUSSELL; NORVIG, 2010)

De acordo com Russell e Norvig (2010) existem três tipos de aprendizado que diferem no tipo de *feedback* informado ao agente que busca se aprimorar. No aprendizado não-supervisionado o agente busca aprender padrões nas entradas de dados sem que qualquer tipo de *feedback* seja informado de maneira explícita. O tipo mais comum de atividade deste tipo de aprendizado seria a criação de grupos a partir de entradas, por exemplo, um cachorro pode desenvolver uma noção sobre dias de passear ou dias de ficar em casa de acordo com o clima do dia sem que o seu dono tenha explicitamente dito que era a hora de passear.

No aprendizado por reforço o agente aprende a partir de incentivos ou punições que são aplicados de acordo com o resultado da predição de saídas a partir de novas entradas. Um exemplo deste tipo de aprendizado seria quando o dono de um cachorro tenta ensinar novos truques a ele oferecendo um biscoito quando ele realiza o truque de maneira correta,

e falando com uma voz mais firme sem oferecer o biscoito quando ele realiza o truque de maneira incorreta. (RUSSELL; NORVIG, 2010)

No aprendizado supervisionado o agente observa pares de entradas e saídas e tenta criar novas funções para reproduzir estes resultados a partir de novas entradas. Um exemplo deste tipo de aprendizado acontece quando uma criança é ensinada sobre como realizar cálculos matemáticos simples e o professor apresenta exemplos de cálculos corretos esperando que a criança realize o mesmo cálculo com valores diferentes e resultados diferentes por si só, sendo informada se o resultado estava correto ou não. (RUSSELL; NORVIG, 2010)

Um exemplo da aplicabilidade de algoritmos de aprendizado de máquina seria o reconhecimento de escrita. Devido a grande variedade de tipos de escrita, este problema conta com um alto grau de complexidade de implementação. Para este problema é utilizada uma imagem de 28 por 28 pixels como representação de um caractere escrito a mão, que é apresentada para o algoritmo como um vetor composto por 784 números reais. A criação de um algoritmo utilizando heurísticas e regras feitas para casos específicos acabaria criando um grande volume de regras e exceções o que levaria a resultados ruins no reconhecimento. (BISHOP, 2006)

Resultados melhores podem ser atingidos utilizando algoritmos de aprendizado de máquina. Um grande volume de dígitos escritos a mão representados por um conjunto de vetores e um vetor com valores representando categorias manualmente escolhidas por um desenvolvedor são conhecidos como *training sets*, e são utilizados como referência para a realização de ajustes nos parâmetros de um algoritmo de aprendizado de máquina. A ideia por trás deste método é que o algoritmo passe a categorizar novos vetores que não fazem parte do *training set* original nas categorias definidas previamente. A habilidade de reconhecer estes novos padrões corretamente é conhecida como *generalização*. (BISHOP, 2006)

### 3 TRABALHOS CORRELATOS

Neste capítulo serão expostos trabalhos que contam com métodos de captura de movimentos de mão utilizando dispositivos com funções similares à luva Hand.io, para uma melhor comparação os trabalhos selecionados contam com sensores acelerômetros e giroscópios como principal método de captura de sinais, assim como a luva Hand.io. Os métodos dos estudos serão analisados de maneira crítica afim de permitir uma comparação entre os pontos fortes e fracos de cada abordagem em relação a este trabalho.

Os trabalhos demonstrados contam com diferentes abordagens no registro dos gestos sendo eles: *Accelerometer-based gesture control for a design environment* (KELA et al., 2006), que propõe um pequeno dispositivo para controle de um estúdio de design inteligente, *I'm home* (KÜHNEL et al., 2011), que apresenta um método bem detalhado para a definição um dicionário de gestos eficientes utilizando a participação dos usuários durante o período de desenvolvimento do sistema, *uWave* (LIU et al., 2009), que demonstra um algoritmo eficiente de reconhecimento de gestos com uma baixa necessidade de recursos computacionais, e por fim *A survey of glove-based systems and their applications* (DIPIETRO et al., 2008) faz uma análise profunda no campo de dispositivos vestíveis, em específico luvas de controle, mostrando desde os dispositivos mais primitivos até os utilizados atualmente.

Estes trabalhos demonstram a viabilidade do desenvolvimento da Hand.io e sua aplicação em um ambiente real. As principais particularidades da Hand.io seguem listadas abaixo e servem de parâmetro de comparação com as demais aplicações:

- Captura passiva de gestos;
- Sensores presos à mão;
- Aplicado em um ambiente residencial;
- Presença de sensor acelerômetro;
- Presença de sensor giroscópio;
- Conexão sem fio;
- Controle por infravermelho.
- Personalização de gestos.



### 3.1 Accelerometer-based gesture control for a design environment

O artigo escrito por [Kela et al. \(2006\)](#) conta com dois estudos principais, sendo o primeiro a aplicação e o estudo da viabilidade da utilização de gestos em um estúdio de design inteligente, em específico em aplicações de Desenho Assistido por Computador (CAD) e o segundo uma avaliação e comparação da interface por gestos em relação a outras modalidades como: controle por voz, objetos físicos utilizando RFID, uma tela de toque através de um tablet, e um dispositivo conhecido como IntelliPen, que funciona com um laser apontado para uma tela que realiza funções similares à um mouse. Foi utilizado um Modelo Oculto de Markov (HMM) ([RABINER, 1989](#)) a partir de sinais discretos para o reconhecimento dos padrões, o algoritmo conta com duas fases distintas de treino e de reconhecimento.

O dispositivo utilizado para captura de movimentos foi uma *SoapBox* ([TUULARI; YLISAUKKO-OJA, 2002](#)), que é uma placa miniaturizada do tamanho de uma caixa de fósforos que conta com um processador, um giroscópio, um acelerômetro, um compasso eletrônico, e comunicação com e sem fios. O protótipo em questão conta com dois botões que são pressionados no início e no fim da realização de um gesto, além do modo de reconhecimento de gestos o protótipo conta com um modo de leitura contínua de movimentos, que pode ser acionado pressionando ambos os botões. Este modo realiza medições diretas dos valores fornecidos pelos sensores e os utiliza para ações de zoom e rotação de objetos virtuais 3D, permitindo que o usuário sinta que está segurando o objeto em suas mãos.

Os resultados do estudo apontam que existe um grande potencial para a utilização de controle por gestos em um ambiente de trabalho, em específico para o aumento da produtividade em determinadas atividades. De acordo com o grau de conhecimento dos participantes dos testes houveram diferentes preferências quanto a interface de utilização do estúdio inteligente, participantes com um grau maior de experiência com design se mostraram mais favoráveis a IntelliPen, devido a similaridade à um mouse, já os demais também demonstraram um interesse maior pela interface por gestos.

O estudo conclui que a utilização de gestos combinados com outros métodos de controle pode ser benéfica para o aumento da produtividade em um ambiente de design, por permitir um controle mais natural e intuitivo de objetos 3D e facilitar a realização de atalhos em programas CAD.

Existem diversos pontos de divergência entre o trabalho apresentado e a Hand.io, sendo o mais predominante o fato dos sensores da Hand.io estarem presos diretamente à mão do usuário, na tabela a baixo as diferenças entre os dois trabalhos são apontadas de forma sucinta.

Table 1 – Comparação entre Hand.io e SoapBox

Recursos	Hand.io	SoapBox
Preso à mão	x	
Utiliza botões		x
Captura contínua de sinais	x	
<b>Sensores</b>		
Acelerômetro	x	x
Giroscópio	x	
<b>Contexto aplicado</b>	Residencial	Estúdio de design

### 3.2 I'm home: Defining and evaluating a gesture set for smart-home control

O artigo *I'm home* de Kühnel et al. (2011) apresenta uma metodologia extensa para definição e avaliação de gestos para controle de ambientes inteligentes, em específico casas inteligentes. O dispositivo escolhido para a realização da captura dos gestos foi um *Smart-phone Iphone*, tal dispositivo foi escolhido devido a presença de um sensor acelerômetro e giroscópio e a sua grande disponibilidade. Os sinais enviados pelos sensores foram processador por um algoritmo Fast Dynamic Time Warping (FastDTW) (SALVADOR; CHAN, 2007) que conta com uma complexidade de tempo de de espaço linear.

Foram realizados três estudos que buscavam definir um vocabulário de gestos simples e coeso, definir o grau de distinção entre diferentes gestos e o grau de dificuldade na memorização de cada gesto.

No primeiro estudo os participantes foram colocados em um quarto com diversos dispositivos de maneira que todos os dispositivos controláveis estivessem no angulo de visão dos voluntários em todos os momentos. As ações realizadas pelo sistema foram realizadas pelos experimentadores através de uma interface gráfica seguidos por uma descrição do comando executado. Os participantes então, foram perguntados qual gesto seria mais apropriado para comando realizado, os movimentos foram gravados com duas câmeras e foram guardados para avaliação do tempo e articulação de cada gesto.

Após a realização de cada gesto foi realizada uma pesquisa para saber a compatibilidade do gesto com a ação realizada e para saber a facilidade da execução do gesto em questão. Foram realizadas 23 ações diferentes que controlavam dispositivos diferentes.

Os resultados deste estudo apontam que a utilização de gestos pré definidos e gestos personalizados são a melhor abordagem no controle por gestos. A utilização de uma grande base de usuários durante o desenvolvimento facilita grandemente a concepção de gestos que possam ser simples e coesos para as atividades propostas. Experiências prévias com interfaces de controle, como a interruptores e reguladores rotativos, podem

influenciar bastante o tipo de gesto escolhido pelos usuários. Foi constatado também que nem o tamanho do gesto nem o tempo necessário para a realização são boas referências para a distinção entre gestos realizados pelos usuários.

No segundo estudo foram exibidas gravações dos gestos realizados no estudo anterior e logo depois foram perguntadas quais as melhores ações poderiam ser realizadas a partir daquele gesto. O ideal seria que os gestos apresentados remetessem aos comandos para os quais eles foram criados. Os gestos que fossem corretamente relacionados às ações mais vezes seriam os mais adequados para a atividade em questão.

Neste estudo foi constatado que gestos mais simples tendem a ser relacionados com mais facilidade com a ação para qual estes foram criados. Movimentos intuitivos como para cima ou para baixo foram relacionados com mais facilidade à atividade que tinham sido propostos, gestos simbólicos como sinais de interrogação também tiveram bons resultados.

No terceiro estudo participantes que não tiveram relação nenhuma com os estudos anteriores. No primeiro momento os voluntários foram convidados a assistir um vídeo para cada gesto e sua respectiva ação. Cada gesto foi repetido 5 vezes por cada participante utilizando um smartphone que registrou os sinais dos gestos que vão ser utilizados como referência para um classificador. Os participantes então, foram perguntados qual o grau de compatibilidade entre o gesto e a ação relacionada a ele.

No segundo momento foram exibidos aos participantes somente as ações que cada gesto realizava em ordem aleatória e durante os 5 segundos seguintes foi requisitado que o gesto correspondente à ação fosse realizado. Caso o gesto fosse correto a apresentação continuaria, caso contrário, o gesto correto seria mostrado e a ação seria exibida ao final da apresentação, que continuaria até que todos os gestos fossem realizados corretamente. Foram registradas a quantidade de erros para cada gesto.

O terceiro estudo conclui que os usuários conseguem distinguir gestos intuitivos de gestos menos intuitivos. Algo que foi comprovado pela quantidade de erros durante o teste de memorabilidade. Gestos mais simples e que rápidos são os mais indicados.

### 3.3 uWave: Accelerometer-based personalized gesture recognition and its applications

O trabalho realizado por [Liu et al. \(2009\)](#) propõe um algoritmo de reconhecimento de gestos chamado *uWave*, que é baseado no algoritmo *Dynamic Time Warping* (DTW) ([SALVADOR; CHAN, 2007](#)), diferentemente de métodos estatísticos como o Modelo Oculto de Markov (HMM) ([RABINER, 1989](#)), que requer um volume muito grande de amostras de referência, o *uWave* necessita de apenas uma amostra para realizar o reconhecimento de um gesto, algo que é muito vantajoso para a criação de gestos personalizados especificamente

para cada usuário. O algoritmo foi criado para ser aplicado em qualquer dispositivo que conte com um sensor acelerômetro de três eixos, tal sensor foi escolhido devido à sua grande disponibilidade em dispositivos eletrônicos de consumo, o que aumenta consideravelmente a aplicabilidade do algoritmo em um ambiente real.

No artigo são realizadas duas aplicações distintas do algoritmo afim de demonstrar a sua versatilidade e usabilidade em cenários reais, a primeira tem como objetivo a identificação simples e precisa de um usuário através de gestos personalizados, analogamente à uma senha alfanumérica. Nesta esta aplicação foi utilizado um controle de um Nintendo Wii como método de entrada de gestos. O estudo constata, através de uma pesquisa realizada com os usuários, que um gesto personalizado é um meio de identificação muito mais fácil de memorizar do que uma senha composta por vários caracteres.

A segunda aplicação é a utilização de gestos para navegação de um ambiente 3D, foi desenvolvida uma rede social baseada em compartilhamento de vídeos utilizando um *smartphone* da Motorola, que conta com um acelerômetro integrado. A aplicação permite que o usuário realize um remapeamento dos gestos para qualquer função específica do aplicativo, o que reforça a natureza de personalização do algoritmo.

Em sua conclusão [Liu et al. \(2009\)](#), discorre sobre os desafios encontrados em definir um vocabulário de gestos que possam ser intuitivos para a grande maioria dos usuários. Outro ponto frisado é o fato do algoritmo depender apenas de um acelerômetro, o que dificulta no registro de certos gestos devido à falta de dados sobre a inclinação e o ângulo das forças realizada sobre o sensor, gerando uma certa confusão nos dados dos movimentos.

Diferentemente do uWave, a Hand.io utiliza um módulo MPU 6050 que conta com um acelerômetro e um giroscópio integrados, o que permite que seja realizada uma medição mais precisa da direção dos movimentos realizados com a luva. Na tabela abaixo estão resumidos os principais pontos de diferenciação deste trabalho em relação ao uWave.

Table 2 – Comparação entre Hand.io e uWave

Recursos	Hand.io	uWave controle	uWave autenticação
Preso à mão	x	x	
Utiliza botões		x	
Captura contínua de sinais	x		
<b>Sensores</b>			
Acelerômetro	x	x	
Giroscópio	x	x	
Compasso eletrônico		x	
<b>Contexto aplicado</b>	Residencial	Controle	Autenticação

adicionar  
refer-  
ência

fazer  
tabela  
de  
com-

## 4 MÉTODO PROPOSTO

Durante este capítulo serão descritas em detalhes as diferentes etapas de funcionamento da luva Hand.io e será explicado como os objetivos deste trabalho serão satisfeitos.

### 4.1 Visão geral do método da Hand.io

A Hand.io é uma luva de controle que permite que quem a use realize gestos para controlar os dispositivos disponíveis em um dado ambiente. O projeto conta com uma luva de pano com um dispositivo preso ao tecido, que lê os movimentos da mão do usuário e uma central de processamento posicionada em um local estratégico, que recebe os sinais dos gestos através de uma conexão sem fio e executa as ações previamente definidas correspondentes a eles.

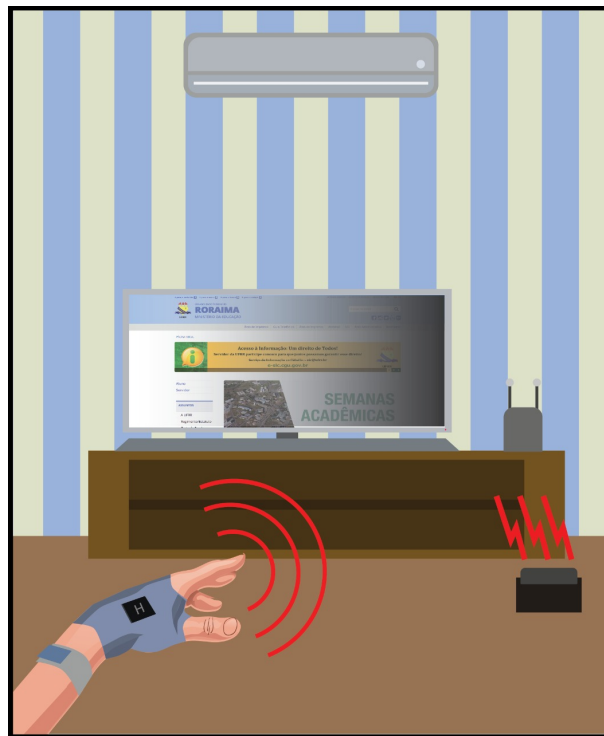


Figure 2 – Visão geral do protótipo

Os sinais dos movimentos são processados pela central utilizando algoritmos de aprendizado de máquina, que classificam o gesto realizado e realizam as ações correspondentes que estão definidas no banco de dados do sistema.

## 4.2 Fluxo de execução da Hand.io

A [Figura 3](#), descreve o método proposto para a luva Hand.io através de um fluxograma, mostrando as diferentes etapas de funcionamento da luva de maneira sequencial.

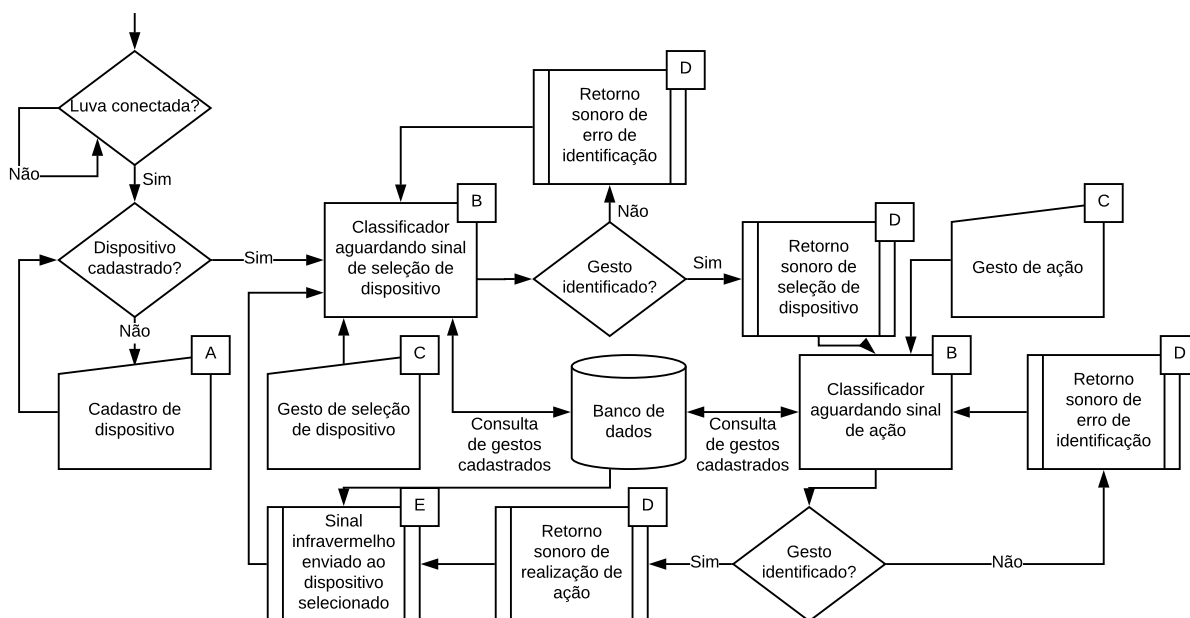


Figure 3 – Fluxograma da Hand.io

- A posição inicial do fluxograma é definida por uma **seta sem origem** a partir da qual são realizadas as checagens iniciais do sistema.
- Os **losangos** representam condições onde dependendo do resultado o fluxo do sistema pode ser alterado, no primeiro momento é verificada a conexão com a luva de controle e logo depois o sistema verifica se existem dispositivos cadastrados, caso não existam o sistema aguarda pela inserção de dispositivos no banco de dados.
- Os **trapézios** representam ações externas que são realizadas pelo usuário, seja a partir da luva ou a partir do dispositivo que será utilizado para realizar o cadastro de novos dispositivos.
- **Retângulos** representam processos internos do sistema e quadrados com barras laterais representam os processos predefinidos do sistema.
- O banco de dados que irá armazenar os dados de gestos e códigos de infra-vermelhos de controle de dispositivos, está representado por um **cilindro** encontrado na parte central da figura.

## 4.3 Captura de Dados Baseado em Movimentos de Amplitude de Punho e Mão

A ideia principal da luva Hand.io é que o usuário controle seu ambiente utilizando gestos da maneira mais natural possível, para isso os movimentos do usuário são capturados constantemente e enviados para a central em tempo real até que seja identificado um gesto correspondente a um dispositivo cadastrado, como pode ser observado na [Figura 3](#).

A luva Hand.io utilizará dois sensores para realizar a captura dos movimentos da mão do usuário, um acelerômetro e um giroscópio. Estes dois sensores foram escolhidos visando evitar problemas os encontrados no trabalho de [Liu et al. \(2009\)](#) encontrado na [section 3.3](#) que sofre problemas de precisão devido a presença de apenas um sensor acelerômetro.

Os sensores escolhidos realizam diversas amostras nas mudanças na aceleração e no giro realizados na luva,

### 4.3.1 Conexão com a central de processamento

A central de processamento inicia a sua operação esperando o recebimento de algum sinal da luva e até que a conexão tenha sido confirmada, não realiza qualquer ação

A conexão entre a luva e a central será realizada utilizando o protocolo de redes sem fio IEEE 802.11 ([CROW et al., 1997](#)), conhecida popularmente como Wi-Fi, através de uma rede LAN. Os sinais

## 4.4 Reconhecimento de Padrões Baseado em Movimentos e Ações

Os sinais recebidos pela central de processamento servem de entrada para algoritmos de aprendizado de máquina que ficam em constante execução tentando classificar os sinais recebidos em categorias previamente escolhidas pelo desenvolvedor do sistema.

Os possíveis gestos reconhecidos pela luva serão definidos durante a fase de implementação e contarão com a ajuda de um grupo de voluntários que servirão de referência para o desenvolvimento de gestos simples e coesos. O trabalho de [Kühnel et al. \(2011\)](#) encontrado na [section 3.2](#) demonstra que um grande grupo de voluntários é fundamental para a criação de um vocabulário de gestos eficiente.

Gestos pré definidos e gestos criados pelos usuários são a abordagem mais efetiva para controlar um ambiente. No entanto uma quantidade muito grande de gestos e gestos muito complexos não são a melhor escolha, pois existe uma quantidade limitada de gestos que podem ser lembrados com precisão sem que haja confusão durante a realização de um determinado gesto.

Será desenvolvido um extensivo *training set* com dados recolhidos de voluntários realizando os gestos. Estes dados serão utilizados como modelo de treinamento para algoritmos de aprendizado de máquina que irão reconhecer os gestos realizados pelo usuário.

Existem dois momentos durante o fluxo de execução da Hand.io onde estes algoritmos são utilizados, quando o sistema aguarda que o usuário selecione o dispositivo que ele deseja controlar e quando o sistema aguarda um comando que será realizado no dispositivo selecionado, ambos os momentos podem ser vistos respectivamente na [Figura 3](#) na letra B. A cada vez que os algoritmos finalizam a classificação dos sinais tidos como entrada, o sistema retorna para o usuário um sinal sonoro de confirmação.

## 4.5 Modelo de Conexão Entre Dispositivos Eletrônicos

O controle dos dispositivos se dará utilizando um LED infravermelho, seguindo um padrão de controle remoto bem estabelecido pela indústria de eletrônicos de consumo. Cada fabricante define códigos infravermelhos específicos para cada dispositivo. Códigos de controle diferentes foram escolhidos visando impedir a interferência entre controles e dispositivos de fabricantes diferentes, como por exemplo o usuário tentar ligar o ar condicionado e acabar ligando a TV por acidente.

Os códigos de controle remoto de cada dispositivo cadastrado no sistema ficarão armazenados em um banco de dados localizado na central de processamento. Cada código estará relacionado à um gesto específico que quando classificado pelo sistema será executado.

## 4.6 Modelo de Prototipação

Os componentes utilizados para a confecção do protótipo da luva e da central buscam ter o menor custo possível, sem comprometer a qualidade do sistema. Componentes produzidos em massa que já tem sua utilização testada extensivamente pela comunidade foram selecionados.

O protótipo é composto pela luva em si que conta com sensores que irão capturar os movimentos do usuário apresentado na [Figura 4](#), e pela central de processamento que irá realizar as ações correspondentes aos gestos exibida na [Figura 5](#), nos dispositivos conectados.

A luva será composta por uma placa GY-521 que tem um sensor MPU-6050, apresentado no canto direito da [Figura 4](#). Esta placa é utilizada em diversos projetos que utilizam captura de movimentos devido a sua boa relação custo benefício em relação a precisão dos sensores. O sensor MPU-6050 conta com um acelerômetro de três eixos e um processador de sinais digitais integrado



A comunicação com a central de processamento ocorrerá utilizando uma placa Wi-Fi ESP-8266

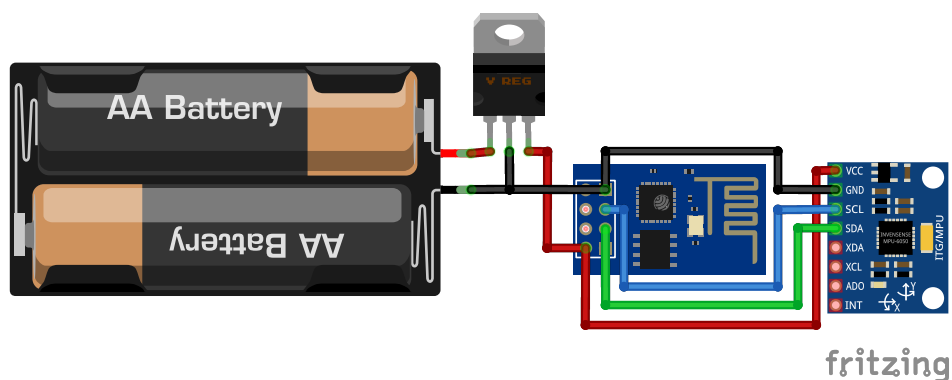


Figure 4 – Esquemático do protótipo da luva

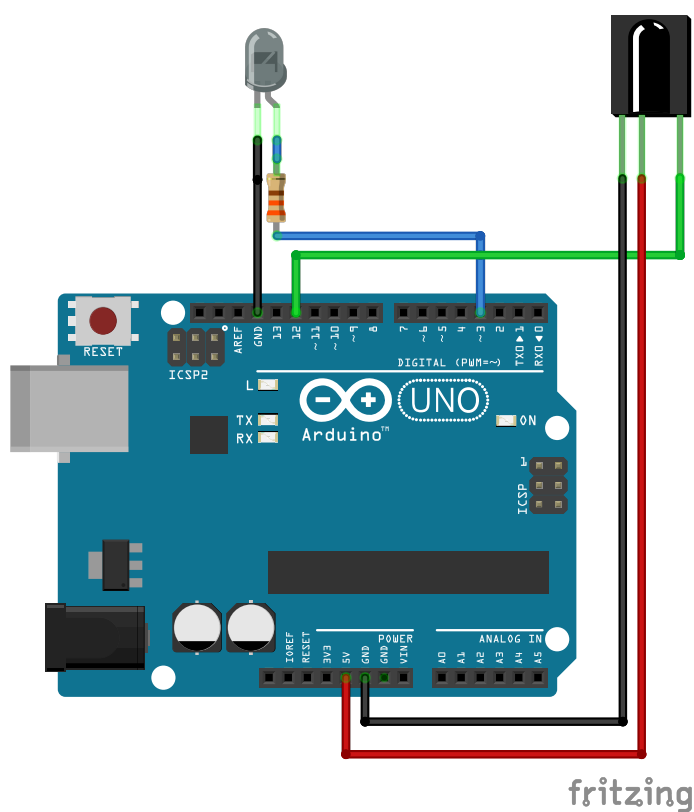


Figure 5 – Esquemático da central de processamento de sinais e execução de ações

## 5 RESULTADOS PARCIAIS

## 6 CONSIDERAÇÕES PARCIAIS E PRÓXIMO PASSOS

# 7 APÊNDICE 1: REVISÃO SISTEMÁTICA DA LITERATURA

## 7.1 Planejamento da Revisão Sistemática da Literatura

Table 3 – Objetivo do estudo.

<b>Analisar</b>	publicações científicas de um estudo baseado em revisão sistemática
<b>Com o propósito de</b>	identifica-las
<b>Em relação às</b>	vantagens e desvantagens da utilização de uma luva inteligente para a captura de movimentos para controle de dispositivos eletro-eletrônicos
<b>Do ponto de vista do</b>	pesquisador
<b>No contexto</b>	acadêmico, industrial, residencial ou urbano para a interação com dispositivos inteligentes ou não.

**Formulação das perguntas.** Buscamos responder as seguintes perguntas:

- Q1: Quais são os principais métodos de captura de movimentos utilizando luvas inteligentes?
  - Q1.1: Qual o sensor ou combinação de sensores que se mostra mais preciso na captura dos movimentos de mão e punho?
  - Q1.2: Qual a melhor disposição de sensores para a confecção da luva?
  - Q1.3: Quais as limitações dos métodos propostos?
- Q2: Qual o algoritmo ou combinação de algoritmos mais eficiente para classificação dos movimentos?
  - Q2.1 Qual o volume suficiente de dados para uma classificação ótima dos movimentos?
- Q3: Qual o melhor protocolo de comunicação entre vestíveis e dispositivos inteligentes em uso atualmente?

**Escopo da Pesquisa.** Para delinear o escopo da pesquisa foram estabelecidos critérios para garantir, de forma equilibrada, a viabilidade da execução (custo, esforço e tempo), acessibilidade aos dados e abrangência do estudo. A pesquisa acontecerá a partir de bibliotecas digitais através das suas respectivas máquinas de busca e, quando os dados não estiverem disponíveis eletronicamente, através de consultas manuais.

**Crítérios de Seleção de Fontes.** Para as bibliotecas digitais é necessário:

- Possuir máquina de busca que permita o uso de expressões lógicas ou mecanismo equivalente;
- Incluir em sua base publicações da área de exatas ou correlatas que possuam relação direta com o tema a ser pesquisado;
- As máquinas de busca deverão permitir a busca no texto completo das publicações.

Além disso, os mecanismos de busca utilizados devem garantir resultados únicos através da busca de um mesmo conjunto de palavras-chaves (string de busca). Quando isto não for possível, deve-se estudar e documentar uma forma de minimizar os potenciais efeitos colaterais desta limitação.

**Métodos de Busca das Publicações.** As fontes digitais foram acessadas via Web, através de expressões de busca pré-estabelecidas. A biblioteca digital consultada foi a Scopus, acessível em <http://www.scopus.com>. Segundo a editora Elsevier (2013), a Scopus é uma das maiores bases de dados de resumos e citações da literatura de pesquisa peer-reviewed com mais de 20,500 títulos de mais de 5,000 editoras internacionais. Dentre estas editoras podemos citar: Springer (Springer (2013)); IEEE Xplore Digital Library (IEEE (2013)); ACM Digital Library (ACM (2013)); ScienceDirect/Elsevier (B.V (2013)); Wiley Online Library (Sons (2013)); British Computer Society (Society (2013)) dentre outras. A biblioteca Scopus também inclui aproximadamente 5.3 milhões de conferências de artigos de proceedings e journals, 400 publicações comerciais, 360 série de livros e publicações aceitas é disponibilizado online antes da publicação oficial em mais de 3,850 periódicos. Ainda segundo a editora Elsevier (2013), a Scopus tem aproximadamente 2 milhões de novas gravações adicionadas a cada ano com atualizações diárias.

**String de Busca.** A string de busca foi definida segundo o padrão PICO (do inglês Population, Intervention, Comparison, Outcomes) (Kitchenham and Charters (2007)), conforme a estrutura abaixo:

- População: Trabalhos publicados em conferências e periódicos que são relacionados com dispositivos vestíveis;
- Intervenção: Relações entre dispositivos vestíveis baseados em capturas de movimentos de mão;
- Comparação: Análise das diferentes abordagens para captura e classificação de movimentos, no sentido de medir a abrangência de cada abordagem diante das métricas propostas não levando em consideração sua eficácia e desempenho, utilizando as questões de pesquisa como fonte para a extração de métricas;

Arrumar  
as  
refer-  
ências  
das  
edi-  
toras.

Arrumar  
as  
refer-  
ências  
da  
string  
de  
busca.

- Resultados: A partir dos relatos das diferentes abordagens de captura de movimentos de mão, pretende-se verificar a eficácia de cada método no contexto de controle de dispositivos eletro-eletrônicos.

Como este estudo representa um estudo de mapeamento/caracterização, a string de busca (para execução na biblioteca digital Scopus como mencionado anteriormente) foi definida de acordo com dois aspectos: População e Intervenção (Peterson et al., 2008), como é apresentado na estrutura abaixo.

- População: publicações que fazem referências à dispositivos baseados em luvas (e sinônimos):
  - "wearable sensors" OR "wearable interaction devices" OR "smart watch" OR "glove-based systems" OR "glove systems" OR "translation glove" OR "unobtrusive wearable" OR "interaction devices" OR "man-machine interfaces"
- Intervenção: captura e análise de movimentos (e sinônimos):
  - "analysis of gestures" OR "hand movement data" OR "gesture recognition" OR "hand gesture interface" OR "hand gesture recognition" OR "gesture recognition framework" OR "gesture-based interaction" OR "personalized gesture" OR "gesture interactions" OR "gesture set"

### 7.1.1 Procedimentos de Seleção e Critérios

A estratégia de busca será aplicada por um pesquisador para identificar as publicações em potencial. A seleção das publicações ocorrerá em 3 etapas:

1. **Seleção e catalogação preliminar dos dados coletados.** A seleção preliminar das publicações será feita a partir da aplicação da expressão de busca às fontes selecionadas. Cada publicação será catalogada em um banco de dados criado especificamente para este fim e armazenada em um repositório para análise posterior;
2. **Seleção de dados relevantes - [1º filtro].** A seleção preliminar com o uso da expressão de busca não garante que todo o material coletado seja útil no contexto da pesquisa, pois a aplicação das expressões de busca é restrita ao aspecto sintático. Dessa forma, após a identificação das publicações através dos mecanismos de buscas, deve-se ler o título, os resumos/abstracts e as palavras-chave e analisá-los seguindo os critérios de inclusão e exclusão identificados a seguir. Neste momento, poder-se-ia classificar as publicações apenas quanto aos critérios de exclusão, entretanto, para facilitar a análise e reduzir o número de publicações das quais se possam ter dúvidas

sobre sua aceitação, deve-se também classificá-las quanto aos critérios de inclusão. Devem ser excluídas as publicações contidas no conjunto preliminar que:

- **CE1-01:** Não serão selecionadas publicações em que as palavras-chave da busca não apareçam no título, resumo e/ou texto da publicação (excluem-se os seguintes campos: as seções de agradecimentos, biografia dos autores, referências bibliográficas e anexas).
- **CE1-02:** Não serão selecionadas publicações em que descrevam e/ou apresentam ‘keynote speeches’, tutoriais, cursos e similares.
- **CE1-03:** Não serão selecionadas publicações em que não se utilize um dispositivo vestível para a captura de movimentos de mão.

Podem ser incluídas apenas as publicações contidas no conjunto preliminar que:

- **CI1-01:** Podem ser selecionadas publicações em que no contexto das palavras-chave utilizadas no artigo levem a crer que a publicação cita um método de captura de movimentos de mão utilizando um dispositivo vestível.
- **CI1-02:** Podem ser selecionadas publicações em que no contexto das palavras-chave utilizadas no artigo levem a crer que a publicação cita diferentes modelos de captura utilizando dispositivos vestíveis recomendando o mais preciso para a aplicação em questão.

3. **Seleção de dados relevantes - [2º filtro].** Apesar de limitar o universo de busca, o 1º filtro empregado não garante que todo o material coletado seja útil no contexto da pesquisa. Por isso, após a leitura na íntegra dos artigos selecionados no 1º filtro, deve-se verificar se as publicações respeitam os critérios abaixo. O objetivo deste 2º filtro é identificar artigos que relacionam a utilização de um dispositivo vestível para a captura de movimentos

- **CS2 -VES -CAP\_MOV** Não devem ser selecionadas publicações que contextualizam dispositivos vestíveis com captura de movimentos.
- **CS2 +VES -CAP\_MOV** Não devem ser selecionadas publicações que citam dispositivos vestíveis, mas não realizam captura de movimentos.
- **CS2 -VES +CAP\_MOV** Não devem ser selecionadas publicações que não utilizem dispositivos vestíveis, mas utilizem captura de movimentos.

Dessa forma, todas as publicações devem respeitar o critério abaixo:

- **CI2 +VES +CAP\_MOV** Só podem ser selecionadas publicações que utilizem dispositivos vestíveis para realizar captura de movimentos.

## BIBLIOGRAPHY

- ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. **Computer Networks**, v. 54, n. 15, p. 2787 – 2805, 2010. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128610001568>. Citado na página 19.
- AYALA, K. J. **The 8051 microcontroller: architecture, programming, and applications**. [S.l.]: West Publishing Co., 1991. Citado 2 vezes nas páginas 17 and 18.
- BENDERS, L. P. M.; STEVENS, M. P. J. Petri net modelling in embedded system design. In: **CompEuro 1992 Proceedings Computer Systems and Software Engineering**. [S.l.: s.n.], 1992. p. 612–617. Citado na página 11.
- BERNIERI, G.; FARAMONDI, L.; PASCUCCHI, F. A low cost smart glove for visually impaired people mobility. In: **2015 23rd Mediterranean Conference on Control and Automation (MED)**. [S.l.: s.n.], 2015. p. 130–135. Citado na página 10.
- BISHOP, C. M. **Pattern Recognition and Machine Learning**. 1st ed. 2006. corr. 2nd printing. ed. [S.l.]: Springer, 2006. (Information science and statistics). ISBN 9780387310732,0387310738. Citado 2 vezes nas páginas 20 and 21.
- BUTTAZZO, G. **Hard real-time computing systems: predictable scheduling algorithms and applications**. [S.l.]: Springer Science & Business Media, 2011. v. 24. Citado 3 vezes nas páginas 6, 16, and 17.
- CHOUDHARY, T.; KULKARNI, S.; REDDY, P. A braille-based mobile communication and translation glove for deaf-blind people. In: **2015 International Conference on Pervasive Computing (ICPC)**. [S.l.: s.n.], 2015. p. 1–4. Citado na página 10.
- CROW, B. P. et al. Ieee 802.11 wireless local area networks. **IEEE Communications Magazine**, v. 35, n. 9, p. 116–126, Sept 1997. ISSN 0163-6804. Citado na página 29.
- CUNHA, E. et al. Formal verification of uml sequence diagrams in the embedded systems context. In: **2011 Brazilian Symposium on Computing System Engineering**. [S.l.: s.n.], 2011. p. 39–45. ISSN 2324-7886. Citado na página 11.
- DIPIETRO, L.; SABATINI, A. M.; DARIO, P. A survey of glove-based systems and their applications. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 38, n. 4, p. 461–482, July 2008. ISSN 1094-6977. Citado na página 22.
- EDWARDS, S. et al. Design of embedded systems: formal models, validation, and synthesis. **Proceedings of the IEEE**, v. 85, n. 3, p. 366–390, Mar 1997. ISSN 0018-9219. Citado 2 vezes nas páginas 18 and 19.
- GIRAULT, C.; VALK, R. **Petri Nets for Systems Engineering**. Springer, 2002. ISBN 3540412174. Disponível em: <https://www.amazon.com/Petri-Systems-Engineering-Claude-Girault/dp/3540412174%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechie-20%26linkCode%3Dxm2%26camp%3D>



3D2025%26creative%3D165953%26creativeASIN%3D3540412174>. Citado 2 vezes nas páginas 11 and 19.

HOPCROFT, J. E.; MOTWANI, R.; ULLMAN, J. D. **Introduction to automata theory, languages, and computation**. 2nd ed. ed. [S.l.]: Addison-Wesley, 2001. ISBN 9780201441246,0201441241. Citado na página 19.

KELA, J. et al. Accelerometer-based gesture control for a design environment. **Personal and Ubiquitous Computing**, v. 10, n. 5, p. 285–299, Aug 2006. ISSN 1617-4917. Disponível em: <<https://doi.org/10.1007/s00779-005-0033-8>>. Citado 2 vezes nas páginas 22 and 23.

KÜHNEL, C. et al. I'm home: Defining and evaluating a gesture set for smart-home control. **International Journal of Human-Computer Studies**, v. 69, n. 11, p. 693 – 704, 2011. ISSN 1071-5819. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1071581911000668>>. Citado 3 vezes nas páginas 22, 24, and 29.

LAMPKA, K.; PERATHONER, S.; THIELE, L. Analytic real-time analysis and timed automata: A hybrid method for analyzing embedded real-time systems. In: **Proceedings of the Seventh ACM International Conference on Embedded Software**. New York, NY, USA: ACM, 2009. (EMSOFT '09), p. 107–116. ISBN 978-1-60558-627-4. Disponível em: <<http://doi.acm.org/10.1145/1629335.1629351>>. Citado na página 11.

LIU, J. et al. uwave: Accelerometer-based personalized gesture recognition and its applications. In: **2009 IEEE International Conference on Pervasive Computing and Communications**. [S.l.: s.n.], 2009. p. 1–9. Citado 4 vezes nas páginas 22, 25, 26, and 29.

MARWEDEL, P. **Embedded system design: embedded systems foundations of cyber-physical systems**. [S.l.]: Springer, 2011. Citado 3 vezes nas páginas 14, 15, and 16.

MILES, K. H. R. **Learning UML 2.0**. 1. ed. [S.l.]: O'Reilly Media, 2006. ISBN 0596009828,9780596009823. Citado 2 vezes nas páginas 18 and 19.

NAVAS, V. X. et al. Smart glove. In: **2012 IEEE Long Island Systems, Applications and Technology Conference (LISAT)**. [S.l.: s.n.], 2012. p. 1–4. Citado na página 10.

O'FLYNN, B. et al. Novel smart sensor glove for arthritis rehabilitation. In: **2013 IEEE International Conference on Body Sensor Networks**. [S.l.: s.n.], 2013. p. 1–6. ISSN 2376-8886. Citado 2 vezes nas páginas 10 and 11.

RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. **Proceedings of the IEEE**, v. 77, n. 2, p. 257–286, Feb 1989. ISSN 0018-9219. Citado 2 vezes nas páginas 23 and 25.

RAMESH, U. B. K.; SENTILLES, S.; CRNKOVIC, I. Energy management in embedded systems: Towards a taxonomy. In: **2012 First International Workshop on Green and Sustainable Software (GREENS)**. [S.l.: s.n.], 2012. p. 41–44. Citado na página 10.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3rd. ed. [S.l.]: Prentice Hall, 2010. (Prentice Hall Series in Artificial Intelligence). ISBN 0136042597,9780136042594. Citado 2 vezes nas páginas 20 and 21.

SALVADOR, S.; CHAN, P. Toward accurate dynamic time warping in linear time and space. **Intell. Data Anal.**, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 11, n. 5, p. 561–580, out. 2007. ISSN 1088-467X. Disponível em: <http://dl.acm.org/citation.cfm?id=1367985.1367993>>. Citado 2 vezes nas páginas 24 and 25.

TUULARI, E.; YLISAUKKO-OJA, A. Soapbox: A platform for ubiquitous computing research and applications. In: MATTERN, F.; NAGHSHINEH, M. (Ed.). **Pervasive Computing**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 125–138. ISBN 978-3-540-45866-1. Citado na página 23.

VAHID, F.; GIVARGIS, T. D. **Embedded System Design: A Unified Hardware/Software an Introduction**. Wiley, 2001. ISBN 0471386782. Disponível em: <https://www.amazon.com/Embedded-System-Design-Hardware-Introduction/dp/0471386782%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0471386782>>. Citado na página 14.

VIVE Chipsets. 2017. Disponível em: <https://www.qualcomm.com/products/vive/chipsets>>. Citado na página 14.

WEISER, M. The computer for the 21st century. **SIGMOBILE Mob. Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 3, n. 3, p. 3–11, jul. 1999. ISSN 1559-1662. Disponível em: <http://doi.acm.org/10.1145/329124.329126>>. Citado na página 11.

WONG, W. E. et al. Recent catastrophic accidents: Investigating how software was responsible. In: **2010 Fourth International Conference on Secure Software Integration and Reliability Improvement**. [S.l.: s.n.], 2010. p. 14–22. Citado na página 11.

YURISH, S. **Advances in Sensors: Reviews, Vol. 3**. Ifsa Publishing, 2016. ISBN 8460877043. Disponível em: <https://www.amazon.com/Advances-Sensors-Reviews-Vol-3/dp/8460877043>>. Citado na página 10.