

# Rede Bayesiana de Emoções: Análise de Usabilidade de um Jogo Digital

Carlos A.S.N.S. Longo, Rafael M. Bulsing, Miguel J. Z. da Costa Junior, Gustavo C. Fleck e Diana F. Adamatti.

**Resumo**— Este artigo tem como objetivo apresentar a proposta de integração entre a placa Arduino, dispositivos acoplados, rede Bayesiana de Emoções e um jogo digital, associando o estudo de emoções com alterações fisiológicas. Algumas ferramentas geradas por este trabalho foram: uma pulseira eletrônica baseada em Arduino capaz de detectar determinados sinais fisiológicos do usuário, como batimentos cardíacos, temperatura e suor, permitindo trabalhos futuros com a finalidade de acompanhar como as mudanças de usabilidade em jogos afetam o usuário no âmbito físico e emocional. Um protótipo de jogo, com sua mecânica e interface determinados, que são o ponto de partida para novos estudos relacionando a usabilidade em jogos as emoções dos jogadores.

**Palavras Chave**— Rede Bayesiana de Emoções, Modelo OCC, Jogo Digital, Arduino.

## I. INTRODUÇÃO

A emoção humana é objeto de estudo em inúmeras áreas do conhecimento sendo o meio computacional uma delas. A Inteligência Artificial (IA) é uma área multidisciplinar que visa a simulação da capacidade humana de pensar, tomar decisões, resolver problemas e nesse caso, sentir. Sistemas Multiagente (SMA) nos oferecem a possibilidade de simular essas situações através de interações entre agentes em um meio comum [1].

Emoções geram reações fisiológicas nos indivíduos, tem um forte caráter subjetivo e tem grande impacto na tomada de ações, decisões e escolhas. Há interesse em simulá-las em ambientes computacionais, utilizando SMA a fim de verificar como emoções estão ligadas com fatores externos. [2]

De mesma forma como nas organizações humanas as

R. M. Bulsing, Carlos A. Longo e Gustavo C. Fleck são estudantes do curso de Engenharia de Computação, do Centro de Ciências Computacionais - C3, da Universidade Federal do Rio Grande - FURG Grande, RS, Brasil (e-mail: rafabulsing@gmail.com, carlos.longo@yahoo.com.br e gustavofleck@furg.br)

Miguel J. Z. da Costa Junior é mestrando do Programa de Pós-Graduação em Computação – Mestrado em Engenharia de Computação, do Centro de Ciências Computacionais – C3 da Universidade Federal do Rio Grande – FURG. (email: mzinelli95@gmail.com).

Diana F. Adamatti é Professora Doutora do Centro de Ciências Computacionais C3 – na Universidade Federal do Rio Grande – FURG.

atividades, muitas vezes, são realizadas por um grupo de decisões individuais que afetam o grupo, em SMA, as pessoas são representadas por agentes artificiais, os quais se relacionam em um ambiente de forma a buscar soluções para problemas de forma cooperativa, compartilhando informações, evitando conflitos e coordenando a execução de atividades [9].

A Rede Bayesiana é uma ótima ferramenta a ser usada no caso da simulação de emoções, pois oferece um raciocínio probabilístico onde podemos adicionar a imprevisibilidade necessária ao agente e assim obter um resultado mais plausível [3]. Silva et al [4] desenvolveram a rede bayesiana de emoções tendo como base no modelo OCC [5].

A placa Arduino é uma plataforma ideal para a criação de protótipos eletrônicos, pois seu microcontrolador pode ser programado através de uma linguagem própria baseada em C/C++, e por existir uma grande variedade de sensores e atuadores compatíveis disponível no mercado [6]. Além disso, o Arduino é uma plataforma gratuita e aberta.

Este artigo tem como objetivo apresentar a integração do Arduino e seus respectivos sensores com um jogo digital e as tecnologias utilizadas para essa integração. Uma das tecnologias, a rede Bayesiana de Emoções, possibilita o desenvolvimento de jogo computacional com uma rede implementada, que recebe sinais fisiológicos do jogador durante sua interação com o jogo, e indiretamente modifica o comportamento do jogo.

## II. REFERENCIAL TEÓRICO METODOLÓGICO

As pessoas que trabalham de modo cooperativo, onde existem O referencial teórico metodológico de um trabalho consiste em apresentar e detalhar os conceitos fundamentais para o entendimento do tema. Nas seções a seguir, são apresentadas as tecnologias utilizadas, os componentes específicos usados para montar o protótipo e o ambiente no qual o estudo de caso será aplicado.

### A) Sistemas Multiagente

Um agente, é definido como um sistema computacional que está situado em algum ambiente, sendo capaz de agir com autonomia neste meio, a fim de atender seus objetivos [11]. Por conseguinte, agentes são estruturas simples que tem uma percepção do ambiente em que ou no qual está localizado, decisões para fazer e a ação para executar alguma tarefa.

Algumas características se destacam quando há o desejo de desenvolver um agente computacional: inteligência, autonomia, capacidade de aprender e cooperar. Existe a necessidade de agentes operarem em sistemas complexos,

dinâmicos e robustos. Segundo Coppin [12], SMA são uma maneira de explorar as características de vários agentes distintos, combinando-os dentro de um sistema computacional.

Costa e Simões [13] definem SMA como sendo composto por vários agentes da mesma classe ou de classes diferentes, que se comunicam entre si direta ou indiretamente, negociando, for-

mando alianças e estabelecendo tipos de interação.

Algumas vantagens do paradigma multiagente dizem respeito a facilidade para modelar domínios distribuídos e ou mundos abertos, além da eficiência para desenvolver quando não há uma solução centralizada e a tolerância a falhas.

### B) Rede Bayesiana

Redes Bayesianas são uma metodologia padrão para a construção dos sistemas que confiam no conhecimento probabilístico, utilizadas em situações que o escopo do sistema é desconhecido.

Matematicamente, uma rede Bayesiana consiste em uma representação compacta de uma tabela de conjunção de probabilidades do universo do problema, combinando conceitos da teoria dos grafos e teoria de probabilidades.

Tendo em vista as dificuldades encontradas para modelagem de problemas reais, tais como a falta de dados e imprevisibilidade dos resultados, optou-se pelo uso de uma Rede Bayesiana.

As redes Bayesianas foram desenvolvidas no início dos anos 1980 para facilitar a tarefa de predição e “abdução” em sistemas de inteligência artificial [3]. Em resumo, essas redes são modelos gráficos para raciocínio baseado na incerteza.

O tipo de probabilidade utilizado nas redes Bayesianas é o condicional. Esse tipo de probabilidade é o que depende de acontecimentos anteriores, ou seja, a probabilidade de um evento acontecer depende dos eventos anteriores. Representa-se por  $P(A|B)$ .

Graficamente são modelos para raciocínio, baseados no conhecimento incerto e incompleto que utilizam dois componentes básicos: nós e arestas. Os nós representam as variáveis, que podem ser discretas ou contínuas; e as arestas assumem as probabilidades de ocorrer o evento.

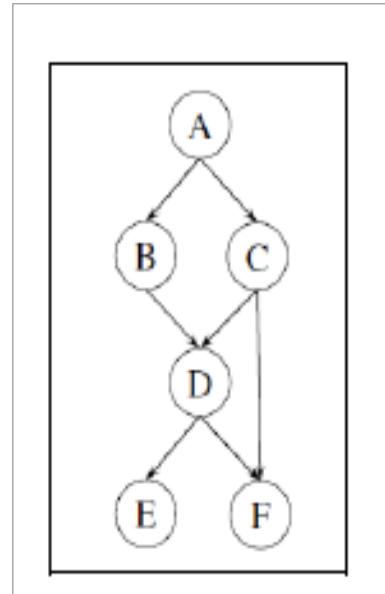


Fig. 1. Rede Bayesiana composta pelos nós  $N=\{A, B, C, D, E, F\}$  e pelas arestas  $E=\{(A, B), (A, C), (B, D), (C, D), (D, E), (D, F), (C, F)\}$  [5]

### C) Emoções

Emoções tem um grande impacto na tomada de decisões, ações, memória e atenção dos indivíduos. Emoções são experiências subjetivas e fortemente associadas a temperamento, personalidade e motivação dos indivíduos. [10]

Para Gratch; Marsella [10] emoções são as regras centrais na vida de humanos. O estado emocional tem grande impacto na tomada de decisões, ação, memória e atenção dos indivíduos.

Existe um grande interesse em simulá-las em ambientes computacionais, para que cenários do cotidiano possam ser estudados em ambientes controlados. Porém, emoções envolvem uma série de complexos fatores sociais e fisiológicos que devem ser observados no estudo desse tema.

### D) Modelo OCC de Emoções

O Modelo OCC de emoções foi proposto no livro “*The Cognitive Structure of Emotions*” por Ortony, Clore e Collins [6]. Esse modelo é capaz de identificar a partir de estímulos gerados pelo ambiente de simulação, as emoções que serão sentidas pelo agente.

Ele usa três tipos de estímulos principais: eventos, agentes e objetos. A emoção sentida é resultado de um ou mais estímulos. O modelo é composto por 22 emoções, como ilustra a figura 2, e se baseia na diferenciação das reações positivas e negativas, ou seja, a partir de um evento, variáveis são atribuídas a fim de gerar uma emoção de negativa ou positiva.

Vale ressaltar que toda emoção gerada no modelo é uma reação a um ou mais aspectos do ambiente. Em indivíduos distintos, um estímulo pode ativar diferentes emoções. Essa divergência ocorre a partir da atribuição de um valor, seja um valor positivo ou negativo, como reação do indivíduo para uma determinada instância.

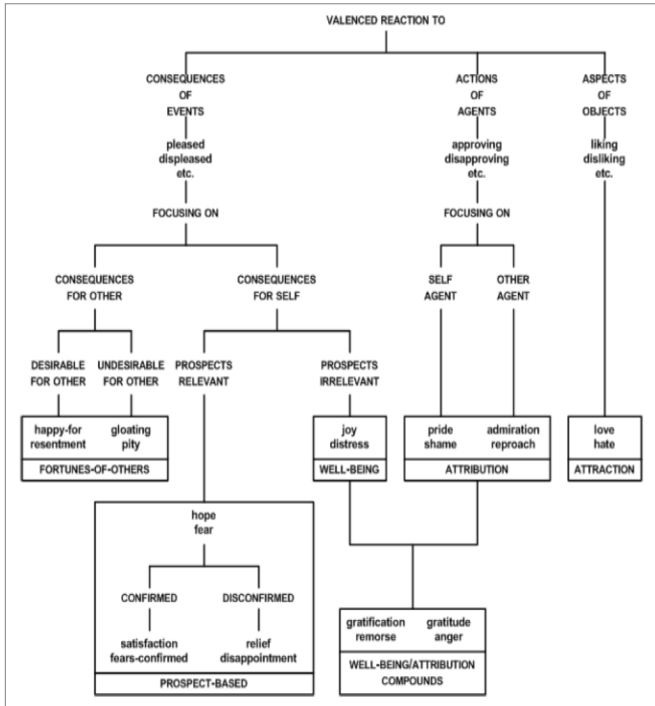


Fig. 2. Estrutura do Modelo OCC [6]

Na parte inferior deste modelo, dentro de retângulos, são apresentadas as 22 emoções com sua nomenclatura em inglês: *Happy-for, Resentment, Gloating, Pity, Hope, Fear, Joy, Distress, Satisfaction, Disappointment, Fears-confirmed, Relief, Pride, Shame, Admiration, Reproach, Love, Hate Gratification, Remorse, Gratitude e Anger*

Em tradução livre: Felicidade, Ressentimento, Regozijo, Pena, Esperança, Medo, Alegria, Estresse, Satisfação, Desapontamento, Medo-confirmado, Alívio, Orgulho, Vergonha, Admiração, Repreensão, Amor, Ódio, Gratificação, Remorso, Gratidão e Raiva.

A estrutura do modelo é apresentada em três grupos importantes, onde cada grupo corresponde a um tipo de estímulo gerado pelo ambiente:

A ramificação localizada mais à esquerda na figura refere-se ao desenvolvimento de emoções geradas em consequências de eventos; enquanto a ramificação central é representada pelas emoções geradas por ações; já a ramificação mais à direita refere-se às emoções produzidas a partir de objetos.

Vale ressaltar que a estrutura do modelo proporciona uma descrição lógica para geração de emoções e dispensa uma sequência temporal.

E) *Modelo de Rede Bayesiana de Emoções*

O modelo da Rede Bayesiana de Emoções foi proposto por Silva; Adamatti e Werhli [4], conforme a figura 3. Este modelo é baseado no modelo OCC. A construção, visualização e análise da rede Bayesiana é feita em Java, através do *software* JavaBayes. O *software* permite a construção de redes com várias estruturas, número indefinido de nós, arestas e variáveis. O resultado desse trabalho é uma rede constituída de 34 nós e 49 arestas.

Cada nó da rede representa um estímulo ou emoção do modelo OCC, com isso é possível introduzir a rede no ambiente

adicionando a imprevisibilidade necessária para a simulação dos eventos.

Na parte inferior da rede Bayesiana encontram-se os nós que representam cada uma das 22 emoções modeladas, fiel ao Modelo OCC.

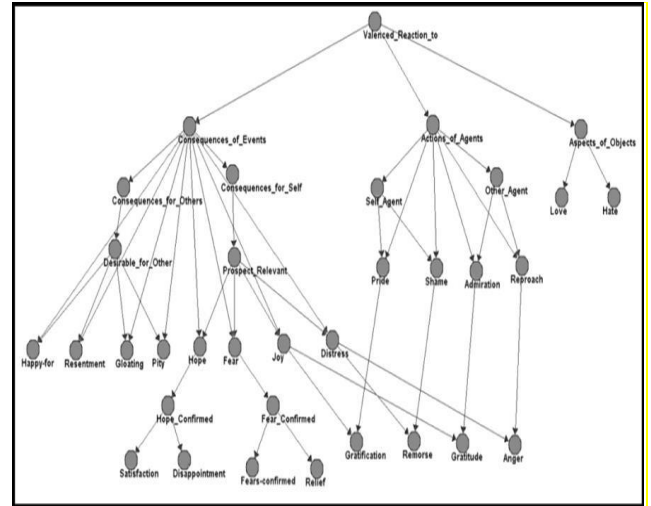


Fig. 3. Rede Bayesiana de emoções [4].

F) *Arduíno*

A placa Arduíno é uma plataforma ideal para a criação de protótipos eletrônicos, pois seu microcontrolador pode ser programado através de uma linguagem própria baseada em C/C++, e por existir uma grande variedade de sensores e atuadores compatíveis disponível no mercado. [7]

Essa plataforma é capaz de ler entradas de diversas formas – luz em um sensor e transformar em uma saída - acendendo um LED ou publicando um conteúdo online. Além disso, é possível informar ao dispositivo o que fazer, enviando um conjunto de instruções para o microcontrolador. Para fazer isso, se usa a linguagem de programação Arduíno (baseada em *Wiring2*) e o *software* Arduíno (IDE) - com base no processamento.

Aplicados a este trabalho, para a coleta dos sinais fisiológicos, o Arduíno foi conectado a sensores de batimentos cardíacos Pulse Sensor SEN-11574 e KY-039, um sensor de temperatura MLX90614 e um sensor de suor Grove GSR.

Ambos os sensores de batimentos cardíacos funcionam emitindo uma luz na pele do usuário, e medindo as pequenas variações causadas pelo fluxo do sangue na corrente sanguínea.



Fig. 4. Pulse Sensor SEN-11574



Fig. 5. Sensor de Batimentos KY-039

O sensor de temperatura, por sua vez, mede a radiação infravermelha emitida pelo jogador, e converte essa informação para a temperatura correspondente em graus *Celsius* ou *Fahrenheit*.



Fig. 5. Sensor de temperatura MLX90614

O sensor galvânico de suor mede diferenças na condutividade elétrica da pele geradas pela salinidade do suor humano.

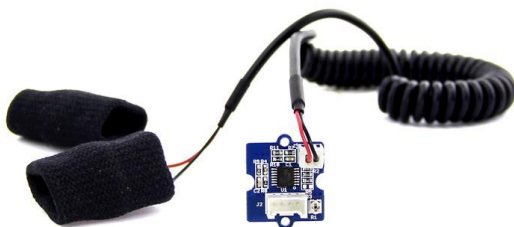


Fig. 6. Sensor de suor Grove GSR

Para enviar os dados coletados pela pulseira para o programa em Java, foi utilizado Javino, que consiste em um protocolo duplo que permite a troca de mensagens entre o Java e o

Arduíno através de uma porta serial. O programa em Java envia uma mensagem à placa pedindo que sejam enviados os dados, e o Arduíno responde enviando as informações que estão sendo captadas pelos sensores no momento [8]. O Javino proporciona detecção de erros *out-of-the-box*, garantindo uma comunicação segura entre o programa Java e o Arduíno.

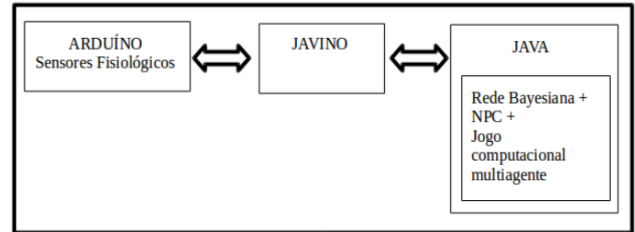


Fig. 7. Estrutura da comunicação entre as partes do modelo.

Os dados recebidos pelo programa Java são sempre no formato *String*, porém esses valores podem então facilmente ser trabalhados e devidamente convertidos para um formato mais apropriado (inteiro, float, entre outros).

O programa do Arduíno foi estruturado de forma que adicionar novos sensores consiste apenas em escrever uma função, o que proporciona facilidade e agilidade para a prototipagem e teste de novos sensores. As Figuras 8 e 9 apresentam um trecho de código que apresenta a codificação no Arduíno e no Java.

```

void loop() {
    if (j.availablemsg()){
        answer(j.getmsg());
    }
}

void answer(String ask){
    if (ask == "temp")
        j.sendmsg(String(mlx.readObjectTempC()));
}
    
```

Fig. 8. Estrutura básica do código Javino no Arduíno.

Da mesma forma, para que o programa Java receba essas novas informações, precisa apenas enviar ao Arduíno uma mensagem com o código para esses dados, fazendo com que essa implementação de novos sensores seja trivial em ambos os lados do projeto.

```

1 import br.pro.turing.javino.*;
2
3 public class Main {
4     public static void main(String args[]) {
5
6         Javino j = new Javino();
7
8         String port = "COM3";
9
10        String ask = "temp";
11
12        j.requestData(port, ask);
13
14        if (j.requestData(port, ask)) {
15            System.out.print("Temp: ");
16            System.out.println(j.getData());
17        }
18    }
19 }
20

```

Fig. 9. Estrutura básica do código Javino no Java.

Assim, tanto o Arduíno quanto o programa Java trabalham em loop infinito. A cada loop, o programa Java envia uma mensagem ao Arduíno, com o código da informação desejada (no exemplo, “temp” para temperatura). O Arduíno, em seu loop, continuamente checa por mensagens, e ao detectar uma, envia a informação relativa ao código recebido.

O modelo de Arduíno utilizado possui uma função que faz com que, quando é estabelecida uma conexão serial, um sinal de Reset é disparado, reiniciando a placa. Isso, porém, impede que o Arduíno se conecte corretamente com o Javino. Para impedir que isso aconteça, foi utilizado um capacitor ligando a porta RESET à uma porta terra GND. O capacitor absorve o sinal elétrico do RESET e, assim, o Javino consegue detectar o Arduíno da forma desejada. A Figura 10 ilustra o erro disparando quando a ligação é feita da maneira incorreta.

```

[[JAVINO] Using version stable 1.1 CEFET/RJ, Brazil
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]
[[JAVINO] Fatal error! [Error on conect COM3]

```

Fig. 10. Saída do console sem o capacitor conectado ao Arduíno

A rede bayesiana, por estar implementada em Java, pode então receber esses dados, e a partir deles, tentar deduzir em tempo real quais emoções e sentimentos o usuário está sentindo no momento da coleta.

Alternativamente, essas informações podem ser também armazenadas diretamente num banco de dados, para que se possa fazer uma análise das mesmas posteriormente.

III. RESULTADOS

A) Jogo

O gênero do jogo é baseado em um *shoot 'em up*. Nesse gênero, o jogador controla um personagem que geralmente é uma espaçonave e tem como objetivo atirar em elementos gráficos dentro do jogo. Este gênero é muito popular e foi originado na década de 80, principalmente em arcades. Um exemplo famoso desse gênero é o jogo *Space Invaders*.

A mecânica do jogo foi idealizada na construção de protótipos de baixa fidelidade, simulando movimento e ações do jogador. Definiu-se a movimentação do personagem apenas em sentido direita-esquerda e uma ação de tiro nos objetos.

Diferente do personagem principal, os inimigos aparecem aleatoriamente no cenário, não tem movimentação. Na Figura 7, é ilustrado um protótipo da interface do jogo. O desenvolvimento do jogo, visando integração com a rede Bayesiana deu-se pela linguagem Java.



Fig. 11 ilustração da interface do jogo.

B) Integração dos componentes

Nesta seção é apresentada a proposta de modelo integrando os componentes apresentados neste artigo, conforme ilustrado na Figura 13.

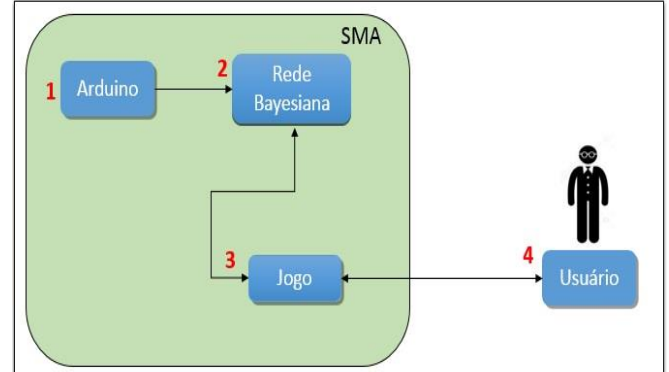


Fig. 12. Modelo de integração dos componentes.

Representado por 1, está a placa do Arduíno e seus sensores

acoplados. Os sensores coletam dados fisiológicos e enviam para a rede Bayesiana de emoções, representada pelo número 2, que armazena pesos para emoções, altera os respectivos pesos da rede e pode também alterar o comportamento do jogo – número 3, liberando mais recompensas ou aumentando a velocidade.

Por conseguinte, o número 4 representa o usuário no papel de jogador, interagindo com os jogos e equipados dos sensores do Arduino para a coleta dos dados fisiológicos.

### C) Coleta dos dados

Os três sensores mencionados (temperatura, suor, batimentos cardíacos) foram já testados em conjunto com o Arduino. Os sensores de temperatura e suor mostram funcionamento satisfatório, apresentando leituras condizentes com a realidade, e esses dados, portanto, podem ser utilizados. A figura 13 faz uma ilustração dos dados coletados pelo sensor de temperatura.

Nenhum dos dois sensores de batimentos cardíacos demonstraram boa performance, porém a leitura de ambos se mostra muito errática, com altas flutuações em tempos muito pequenos sem que o usuário tenha de fato sofrido alguma mudança. Ainda está sendo estudado uma solução para este problema.

```
[[JAVINO] Using version stable 1.1 CEFET/RJ, Brazil
Temp: 14.05
Temp: 14.01
Temp: 14.05
Temp: 13.99
Temp: 14.01
Temp: 14.01
Temp: 14.05
Temp: 14.01
Temp: 13.99
Temp: 13.99
Temp: 14.05
Temp: 14.01
Temp: 14.01
Temp: 14.01
Temp: 14.05
```

Fig. 13. Ilustração dos dados coletados pelo sensor de temperatura.

## IV. CONCLUSÕES E TRABALHOS FUTUROS

O estímulo de emoções pode ocorrer em diversos ambientes e ocasiões, sendo uma característica subjetiva de cada ser humano. A fim de investigar sinais fisiológicos relacionados a emoções, é proposto neste trabalho apresentar os sensores do Arduino, assim como sua integração, funcionamento e resultados preliminares.

Este trabalho inicialmente abre uma série de oportunidades para pesquisa, podendo relacionar jogos, qualidade e usabilidade ou percursos cognitivos com emoções procurando investigar e verificar o comportamento humano em ambientes controlados.

### REFERÊNCIAS

- [1] S. Russel; P. Norvig. *Artificial Intelligence A Modern Approach*. 2a. ed. Upper Saddle River, New Jersey: Prentice Hall, 2003.
- [2] Gratch, J. and Marsella, S. Modeling emotions in the mission rehearsal exercise. 2001. In Proceedings of the 10th Conference on Computer Generated Forces and Behavioral Representation, pages 15–17.
- [3] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. 1a. ed: Morgan Kaufmann, 1988.
- [4] Silva, d. F. N.; Adamatti, D. F.; Werhli, V. A. Using Bayesian Networks to Structure the OCC Emotions Model. *Journal of Intelligent Computing*, [S.l.], p.156–171, 2016.
- [5] A.V.Werhli. *Reconstruction of Gene Regulatory Networks from Postgenomic Data*. Tese de doutorado. University of Edinburgh, 2007.
- [6] A. Ortony; G. Clore; A. Collins. *The Cognitive Structure of Emotions*. Cambridge: Cambridge University Press, 1988.
- [7] Arduino (2017) <http://www.Arduino.cc/>, Acesso em 03 de Fevereiro de 2017.
- [8] Javino (2017) <http://sourceforge.net/projects/javino/>, Acesso em 03 de Fevereiro de 2017.
- [9] D.F.A.Adamatti. *Ambiente de Simulação Baseado em Agentes com Emoções*. Dissertação de Mestrado. Universidade Federal do Rio Grande do Sul. Porto Alegre. 2003.
- [10] Gratch, J.; Marsella, S. Modeling emotions in the mission rehearsal exercise. In: \Conference on Computer Generated Forces and Behavioral representation, 10. p.15–17.
- [11] Wooldridge, Michael, and Nicholas R. Jennings. "Intelligent agents: Theory and practice." *The knowledge engineering review* 10.2 (1995): 115-152.
- [12] Coppin, Ben. *Inteligência artificial*. Grupo Gen-LTC, 2015.
- [13] Costa, Ernesto; Simões, Anabela. *Inteligência artificial: fundamentos e aplicações*. 2004.