

# Lousa Interativa com Software Livre, uma alternativa para as escolas de Volta Redonda

A ideia da Lousa Interativa com software Livre que apresentaremos, surgiu a partir do projeto de Johnny Chung Lee um pesquisador do "Human-Computer Interaction Institute" da "Carnegie Mellon University" nos Estados Unidos.

Além de elaborar o esquema conceitual para construção da Lousa Interativa, Johnny Chung Lee desenvolveu o "Wiimote Whiteboard". Sua ideia inicialmente era de incrementar suas apresentações.

Para tal, desenvolveu um projeto que cria um quadro branco "multipoint interativo" utilizando apenas de:

- Computador;
- Projetor Multimídia;
- Wiimote (Controle sem fio do Nintendo

Wii);

- Adaptador Bluetooth;
- Software de controle;
- Caneta construída com um LED IR e pilhas comuns.

## SITE

Para saber um pouco mais sobre o projeto de Johnny Chung Lee acesse: [link](#)

### Segundo a Wikipédia Quadro Interativo é:

"Um Quadro Interativo é uma superfície que pode reconhecer a escrita eletronicamente e que necessita de um computador para funcionar. Alguns quadros interativos permitem também a interação com a imagem do computador projetada. São geralmente utilizados no escritório e na sala de aula."

### O que é Lousa Interativa ou Quadro Interativo ?

Quadro interativo pode ser considerado qualquer quadro que interage com algo. No entanto o sentido mais correto na área educacional é o de um quadro branco conectado a um computador com um software específico e com projeção da tela sobre ele que interage com as diversas ferramentas disponíveis.

Muitas pessoas também chamam de lousa digital, quadro digital ou ainda Lousa Interativa que é a nomenclatura mais utilizada no Brasil, no entanto

todos estes nomes estão corretos e são basicamente o mesmo sistema de interação.



Figura 1 : Modelo de Lousa Interativa comercial

### Histórico

A partir do projeto original, encontramos alguns problemas para torná-lo viável para o uso em ambiente produtivo, dentre eles:

#### Calibração da lousa

A necessidade de demarcação de pontos onde seria possível a interação entre usuário, lousa e o computador, sendo que para cada ambiente seria necessário refazer um procedimento demorado;

#### Interferência na luz do projetor

O usuário teria que ficar na frente da projeção criando uma situação de desconforto para os expectadores;

## Sempre na frente da luz



Figura 2 : Usuário na frente da apresentação

### Falta de mobilidade

O usuário ficaria preso à lousa (projeção) para conseguir a interação;

### Questão da largura e altura

Quando a altura e a largura da projeção for maior do que o alcance da câmera do "wii" ou em um auditório onde a tela for mais alta que o alcance do operador, seria inviável fazer a calibragem e uso da lousa;

### Controle de wii fixo

Descaracterizaria o potencial de mobilidade do controle e usuário;

### Caneta com LED IR

Só ofereceria a função do botão esquerdo do mouse.

## Esquema de montagem

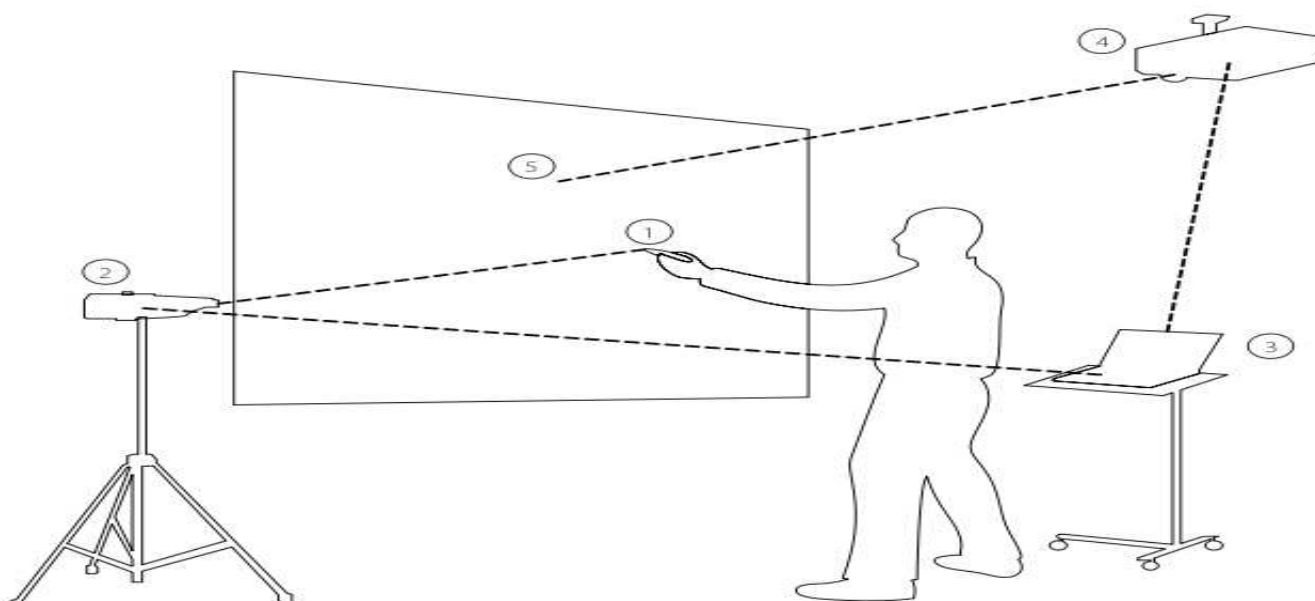


Figura 3 : Montagem

### Nossa Proposta

Após várias horas experimentando as possibilidades e dificuldades do projeto original, verificamos algumas alternativas:

#### Calibração da lousa

Retiramos a necessidade deste procedimento;

#### Interferência da luz do projetor

O usuário não precisa ficar em frente ao projetor;

#### Falta de mobilidade

O usuário pode interagir de qualquer lugar da sala, desde que o sensor ou sensores estejam ao alcance do "wii", uma vez que podemos colocar vários sensores na mesma sala;

#### Controle de wii móvel

Usamos o controle de "wii" no lugar da caneta que possibilita a movimentação e controle do "mouse" em toda tela.

#### Configuramos o controle de wii com as funções:

Botão direito, botão esquerdo, setas direcionais e home, que durante a apresentação volta para página inicial;

#### LEDs IR

Usamos a ideia da caneta para criar sensores que podem ser colocados em posições estratégicas, garantindo a total mobilidade do usuário.

Os laboratórios das Escolas Municipais de Volta Redonda, usam "VRLivre" uma customização do Sistema Operacional "UBUNTU", que foi criada para atender as necessidades do trabalho de Informática Aplicada à Educação e conta com diversos materiais de domínio público, objetos de aprendizagem (RIVED), vídeos e textos disponibilizados pelo "MEC".

Ainda conta com aplicativos para áudio, vídeo, artes gráficas, animação, escritório, programação, internet, multimídia, rede, scripts, programas criados pela coordenação e softwares educacionais de diversas disciplinas.



Figura 4 : Aluna da E.M. Mario Villani utilizando o programa Audacity

Optamos pelo uso do Software Livre, pois, além da redução considerável de custos, proporciona o compartilhamento de saberes, estimulando o trabalho colaborativo, democratizando o acesso às tecnologias e favorecendo o processo de inclusão digital, sendo que o Software Livre nos garante quatro princípios de liberdade:

## Liberdade nº 0

A liberdade de executar o programa, para qualquer propósito



## Liberdade nº 1

A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades. Acesso ao código-fonte é um pré-requisito para esta liberdade.

```
Simple HelloButton() method.  
@version 1.0  
@author john doe <doe.j@example.com>  
HelloButton()  
{  
    JButton hello = new JButton( "Hello, wor  
hello.addActionListener( new HelloBtnList  
// use the JFrame type until support for t  
new component is finished  
JFrame frame = new JFrame( "Hello Button"  
Container pane = frame.getContentPane();  
pane.add( hello );  
frame.pack();  
frame.show(); // display the fra
```



## Liberdade nº 2

A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo.



## Liberdade nº 3

A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie. Acesso ao código-fonte é um pré-requisito para esta liberdade.

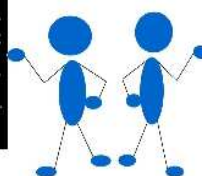


Figura 4 : Liberdades do Software Livre

## Instalando e Configurando os softwares necessários

Você vai precisar de uma conexão com a internet. Abra um terminal, logue-se como "root" e execute os comandos a seguir:

```
# aptitude install bluez-utils  
bluetooth wminput wmgui lswm  
+ enter
```

Entendendo a linha de comando:

- **bluez-utils bluetooth**: softwares para o adaptador bluetooth;
- **wminput wmgui lswm**: softwares para interação e comunicação com o

controle de "wii".

```
root /home/micro01 # aptitude  
install bluez-utils bluetooth  
wminput wmgui lswm
```

## Consertando erros na Instalação

```
# aptitude -f install + enter
```

Para verificar se ocorreu algum erro com dependências durante a instalação e se for necessário fazer os devidos consertos.

Obs.: se os erros persistirem, reveja os processos, procure por erros de digitação nos comandos, veja se a

conexão com a internet está ativa, etc...

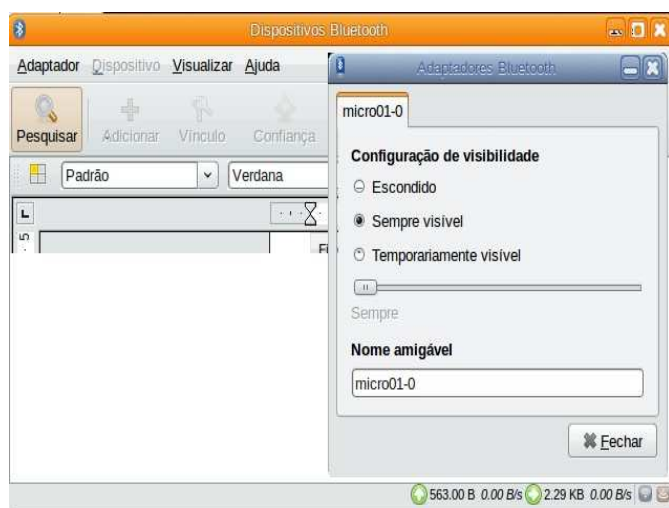
Se tudo correu bem nos passos acima, basta conectar "Adaptador Bluetooth" na porta "USB", que um ícone logo irá aparecer na barra próxima ao relógio.

## Visibilidade do Bluetooth

Precisamos configurar a visibilidade do Adaptador Bluetooth". Para fazer isso, clique no menu

"**Sistema**" > "**Preferências**" > "**Gerenciador Bluetooth**" na tela "**Dispositivos Bluetooth**", vá em

"**Adaptador**" > "**Preferências**" e marque a opção > "**Sempre visível**", clique no botão "**Fechar**".



## Pegando o MAC do controle do wii

**# ls m** + enter

Aparecerá uma mensagem solicitando que você pressione os botões "1 e 2" simultaneamente, do controle de "wii".

Repita o comando acima até que apareça na tela uma sequência de números e letras (endereço "MAC" do controle), indicando que foi encontrado o dispositivo.

Anote e guarde o endereço "MAC" do "wii" que aparecerá. Veja o exemplo na figura abaixo.

```
root /home/micro01 # lswm
Put Wiimotes in discoverable
mode now (press 1+2)...
00:23:31:0D:85:FA
root /home/micro01 #
```

## Carregando o módulo de controle do wii

Adicionando o módulo de controle do "wii" na inicialização do sistema. No terminal digite:

**# gedit / etc/ modules** + enter

No arquivo que será aberto, após a última linha adicione a palavra "**uinput**". Salve e feche o arquivo. Conforme mostra a figura abaixo.

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that
# should be loaded
# at boot time, one per line. Lines beginning with "#" are
# ignored.

lp
lp
lp
lp
lp
lp
lp
lp
lp
lp
lp
uinput
```

## Movimentos do mouse no controle do wii

Vamos dar vida ao mouse. Com estas configurações será possível executar todos os movimentos do mouse apenas movimentando o controle de "wii" na direção em que quisermos levar o cursor. No terminal digite:

```
# cd / etc/ cwiiid/ wminput + enter
```

Para entrar na pasta onde estão os arquivos que serão modificados.

```
# mv default default.bak + enter
```

Para fazer uma cópia de segurança da configuração original.

```
# cp ir_ptr default + enter
```

Para copiar o modelo de configuração.

## Movimentos do mouse no controle do wii

```
# gedit default + enter
```

Para editar o arquivo que vai controlar os movimentos do "wii". No arquivo que será aberto, retire o "~" das palavras "~ ABS\_X" e "~ ABS\_Y" e deixe conforme mostra figura abaixo.

```
#ir_ptr

include buttons

Plugin.ir_ptr.X = ABS_X
Plugin.ir_ptr.Y = ABS_Y
```

Quando terminar, basta salvar e fechar o arquivo.

## Configurando os botões do controle de wii

No terminal digite:

```
# gedit buttons + enter
```

Para abrir o arquivo de mapeamento dos botões de controle do "wii".

Você deve fazer as configurações da forma que estamos sugerindo na imagem abaixo. Desta forma, você terá total controle de movimento e de todas as funções do mouse.

```
#buttons

Wiimote.A = BTN_RIGHT
Wiimote.B = BTN_LEFT
Wiimote.Up = KEY_UP
Wiimote.Down = KEY_DOWN
Wiimote.Left = KEY_LEFT
Wiimote.Right = KEY_RIGHT
Wiimote.Minus = KEY_BACK
Wiimote.Plus = KEY_FORWARD
Wiimote.Home = KEY_HOME
Wiimote.1 = KEY_PROG1
Wiimote.2 = KEY_PROG2

Nunchuk.C = BTN_RIGHT
Nunchuk.Z = BTN_LEFT

Classic.Up = KEY_UP
Classic.Down = KEY_DOWN
Classic.Left = KEY_LEFT
Classic.Right = KEY_RIGHT
Classic.Minus = KEY_BACK
Classic.Plus = KEY_FORWARD
Classic.Home = KEY_HOME
Classic.A = BTN_RIGHT
Classic.B = BTN_LEFT
#Classic.X =
#Classic.Y =
#Classic.ZL =
#Classic.ZR =
#Classic.L =
#Classic.R =
```

## Os comandos do mouse no controle de wii

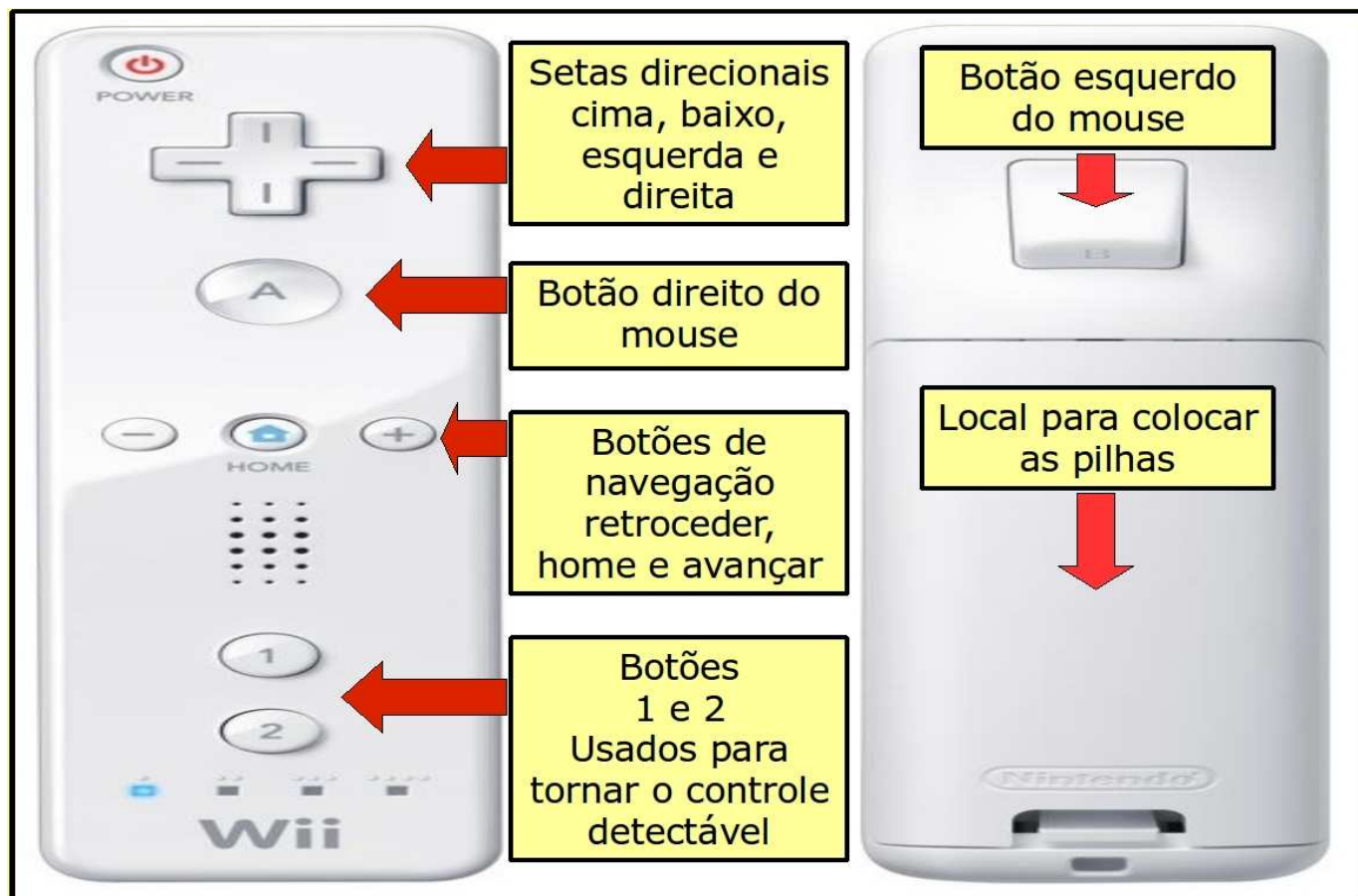


Figura 5 : Configuração utilizada para o controle de wii

### Criando um “script” para automatizar a inicialização do controle de “wii”

```
#!/bin/bash  
# script para iniciar o controle de wii  
wminput 00:23:31:0D:85:FA &
```

No terminal digite:

### Observação

```
# gedit /usr/local/bin/quadro +  
enter  
Para abrir o editor de textos.
```

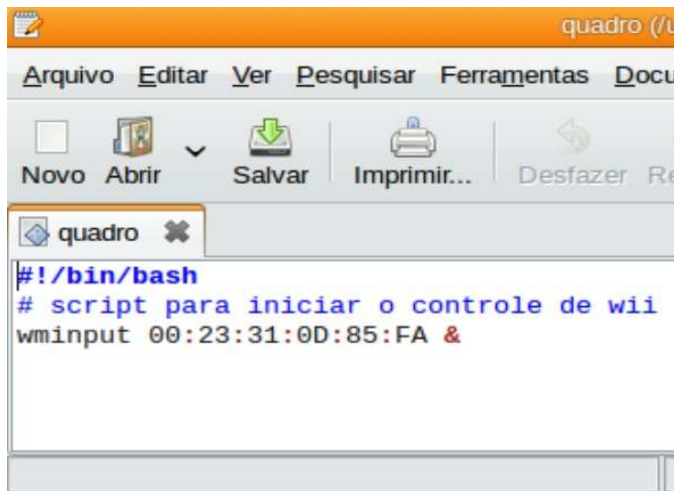
Na tela de edição que se abriu, digite o texto abaixo ou copie e cole.

**Atenção com os caracteres especiais. O texto deverá ser digitado exatamente como estamos mostrando:**

Vale lembrar que no exemplo mostrado, o endereço “MAC” (00:23:31:0D:85:FA), corresponde ao do controle de “wii” que usamos em nossos testes, sendo assim você deverá substituir o mesmo pelo endereço que você conseguiu quando usou o comando “lswm”. Lembre-se de que em cada controle de “wii” tem o seu endereço “MAC” que o torna único dentro do sistema.



Veja como ficou o arquivo, na imagem abaixo.



Quando terminar salve e feche o arquivo.

Agora só precisamos dar permissão de execução para o "script" que acabamos de criar, digite o comando abaixo, no terminal:

```
# chmod +x /usr/local/bin/quadro + enter
```

Para tornar o arquivo executável.

## Reiniciando o computador e finalizando os testes no controle de wii

Se tudo deu certo até aqui, já podemos reiniciar o computador ou digitar no terminal o comando "sudo modprobe uinput + enter" para que o módulo de controle do "wii" (uinput) seja carregado.

Conecte o "Adaptador Bluetooth" e vamos começar o teste final de conexão.

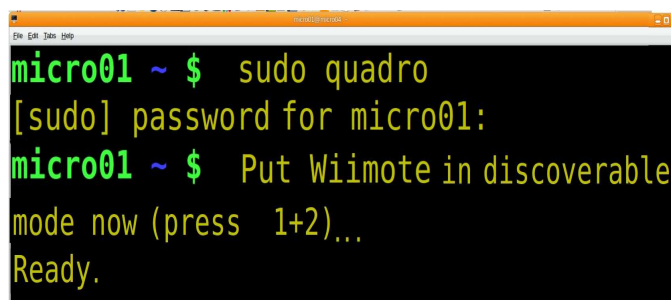
Agora já conseguiremos testar a conexão do "wii" + "Bluetooth" + "Software".

## Conectando o controle de wii + Bluetooth

No terminal, como root, digite:

```
# sudo quadro + enter
```

Será solicitada a senha de administração. Não se esqueça de pressionar os botões "1 e 2" no controle de "wii" para que ele possa ser detectado pelo sistema. Se der erro, repita o procedimento. Quando aparecer a palavra "Ready", como mostra a figura abaixo, significa que você já conseguiu a conexão. Não feche o terminal, apenas deixe a janela minimizada.



(LED IR)

Agora precisaremos montar os sensores que vão permitir que você possa usar o controle de "wii" para movimentar o mouse por toda a tela. Você poderá fazer quantos sensores achar necessário, uma vez que o valor gasto com cada unidade não chega a "R\$ 5,00".

Veja a lista de materiais abaixo:

- um LED IR de 940nm;
- uma pilha “AA” alcalina;
- um interruptor pequeno;
- uma lanterna de bicicleta, encontrada em lojas de R\$: 1,99;
- ferro de solda, pedaços de fio, cola quente, estilete e imaginação.

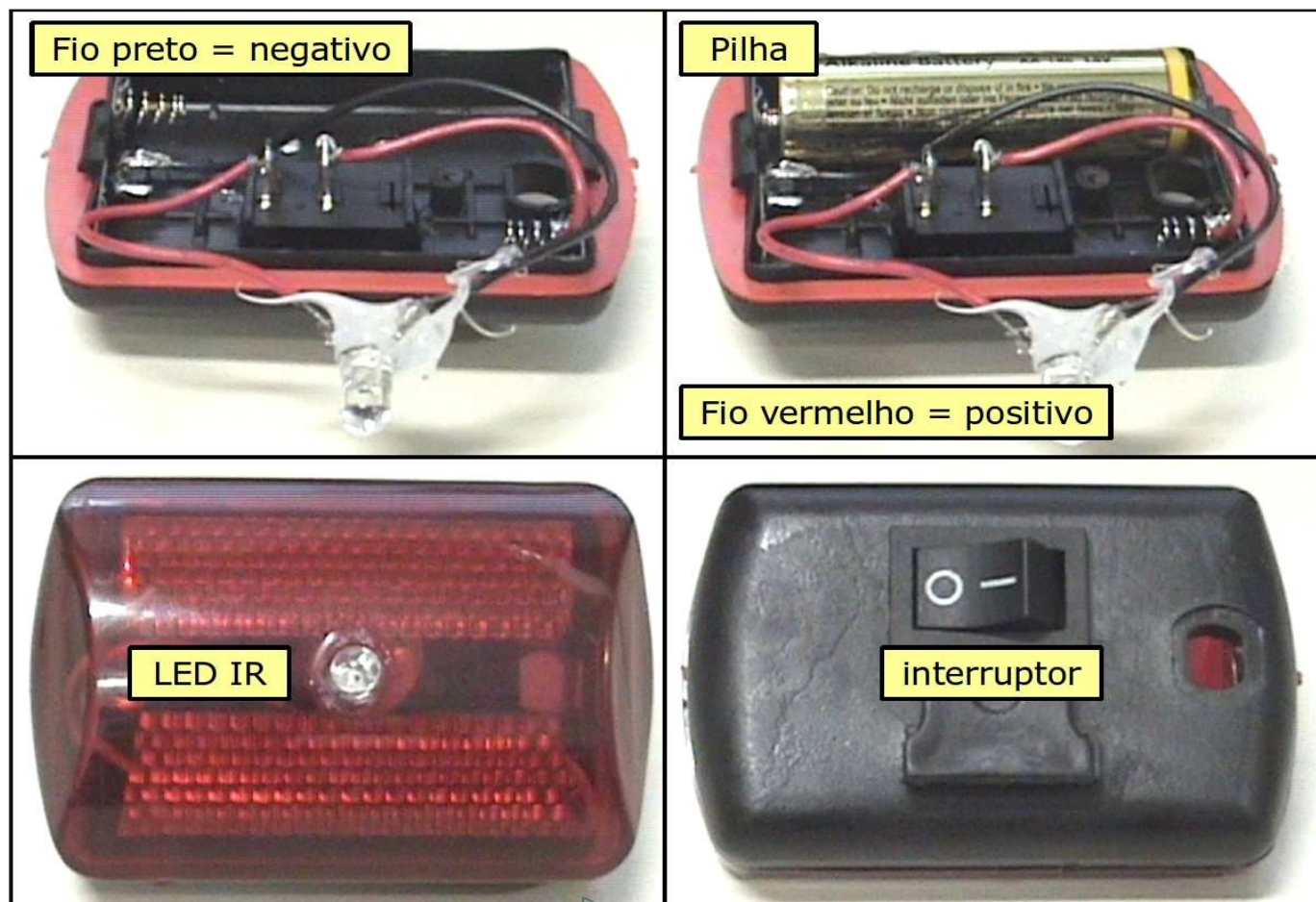


Figura 6 : Montagem do LED IR

## Testando o resultado final do projeto

Para testar o funcionamento do “wii” usando o sensor, basta proceder a conexão do “wii” com o “Bluetooth”, da forma que já foi mencionado anteriormente. Ligue o interruptor do sensor e o coloque a uma distância de aproximadamente um metro e meio do controle de “wii”. Se tudo correu bem

você já vai conseguir movimentar o ponteiro do mouse, fazendo movimentos com o “wii”.

Vale ressaltar que você vai precisar se acostumar com os movimentos do controle, pois o mesmo é muito sensível, ainda mais que até agora você estava acostumado com os movimentos do mouse, que era apoiado em uma superfície plana. Aproveite para testar os botões e as funções do controle.

# PROJETO LOUSA INTERATIVA

Os gastos com materiais usados somam um valor aproximado de  
**R\$ 200,00**



Computador – se já possui o custo é  
**R\$ 0,00 (zero)**



DataShow – se já possui o custo é  
**R\$ 0,00 (zero)**



Controle de wiimote – em média **R\$ 150,00**

kit para construir o sensor infravermelho (solda, pedaços de fio, interruptor, cola quente, estilete, uma lanterna de bicicleta, etc...) - em média **R\$ 10,00**



4 pilhas Alcalinas – em média **R\$ 10,00**



Adaptador Bluetooth – em média **R\$ 30,00**

Para saber mais...

Endereços na web usados como referência durante o desenvolvimento do Projeto Lousa Interativa

- Site do projeto de Johnny Chung Lee  
<http://www.cs.cmu.edu/~johnny/projects/wii>
- Arquivo para download do projeto de Johnny Chung Lee
- Fórum Ubuntu - Como fazer Wii remoto  
<http://ubuntuforums.org/showthread.php?t=836231&highlight=wiimote+karmic>
- Fórum Taringa - Wiimote + Ubuntu 9.10
- Site do projeto Trac
- Site de pesquisa Google
- Vídeos hospedados no Youtube
  - Vídeo 1  
[http://www.youtube.com/watch?v=NwVBzx0LMNQ&feature=player\\_embedded](http://www.youtube.com/watch?v=NwVBzx0LMNQ&feature=player_embedded)
  - Vídeo 2  
<http://www.youtube.com/watch?v=QgKCrGvShZs&feature=related>
  - Vídeo 3  
[http://www.youtube.com/watch?v=5s5EvH7eQ&feature=player\\_embedded](http://www.youtube.com/watch?v=5s5EvH7eQ&feature=player_embedded)



**Sergio Graças** é Coordenador do NTM de Volta Redonda, Desenvolvedor do Projeto GNU/Linux VRlivre, Administrador da Comunidade Linux Educacional no Portal do Software Público Brasileiro, Administrador do Portal IAESMEVR e Técnico de Informática da SMEVR.

<http://www.iaesmevr.org>  
sergio.sme@epdvr.com.br  
Usuário Linux nº 456767  
Usuário Ubuntu nº 25471

**Giany Abreu** é Multiplicadora do NTM e NTE de Volta Redonda, Desenvolvedora do Projeto GNU/Linux VRlivre, Administradora dos Portais IAESMEVR e NTEVR07 e Coordenadora de Informática Aplicada à Educação da SMEVR.

<http://www.iaesmevr.org>  
<http://www.ntevr07.org>  
gianyabreu@yahoo.com.br



**Suellem Oliveira** é Técnica de Informática do NTMVR e SMEVR. Colaboradora do Projeto Lousa Interativa

<http://www.iaesmevr.org>  
suellem.sme@epdvr.com.br

**Theodoro Almeida** é Multiplicador do NTMVR e NTEVR. Colaborador do Projeto Lousa Interativa

<http://www.iaesmevr.org>  
theodoro.sme@epdvr.com.br

