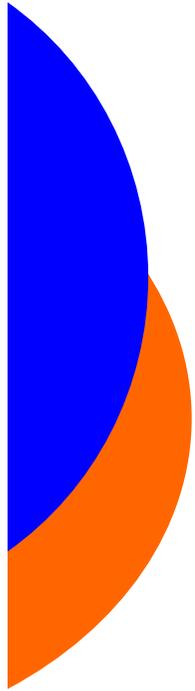


# **PROTÓTIPO DE UM SISTEMA DE SEGURANÇA RESIDENCIAL COM LINUX EMBARCADO**

---

**Acadêmico: Daniel Baumann**

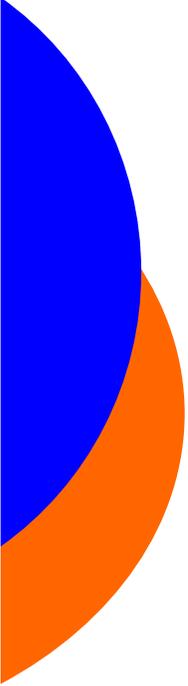
**Orientador: Miguel Alexandre Wisintainer**



# Roteiro

---

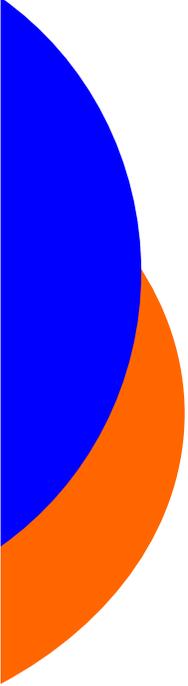
- Introdução.
- Objetivos do trabalho.
- Fundamentação teórica.
- Desenvolvimento do trabalho.
- Especificação de hardware e software.
- Técnicas e ferramentas utilizadas.
- Partes da implementação.
- Estudo de caso.
- Resultados e discussão.
- Conclusão.
- Extensões.



# Introdução

---

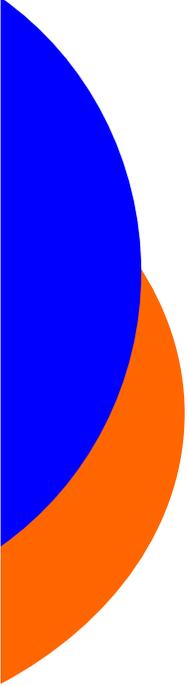
- A preocupação com a segurança residencial é cada vez maior.
- Existem várias maneiras de proteger-se.
- Grande parte dos sistemas de seguranças residenciais possuem monitoramento local e cabeamento específico.
- Surge à idéia de desenvolver um protótipo de sistema de segurança residencial.



# Objetivos do trabalho

---

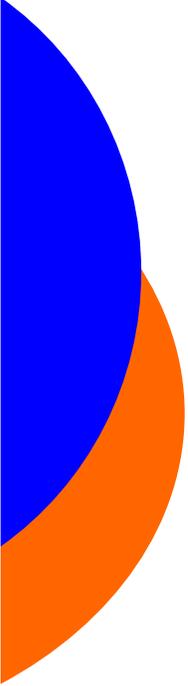
- Criar um servidor web dentro da Fox Board que possibilite o controle de entrada e saída dos pinos de uso geral e visualização de imagem capturadas por uma câmera.
- Capturar imagens de uma câmera acoplada a porta USB da placa Fox Board.
- Construir um circuito eletrônico de potência para acoplar os pinos da placa Fox Board aos sensores, alarme, tomadas e iluminação.



# Fundamentação teórica

---

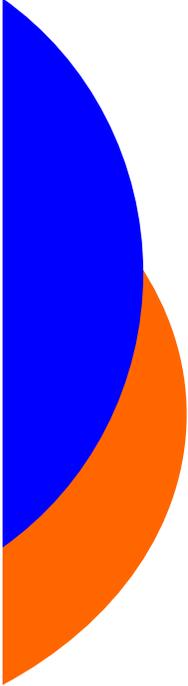
- Conceitos básicos
  - Domótica.
  - CGI.
  - Fox Board.
- Trabalhos correlatos.



# Domótica

---

- O termo domótica é resultado da junção das palavras *domus* (que significa casa) com robótica (que é o controle automatizado de algo).
- Aplicações em domótica:
  - Automação.
  - Iluminação.
  - Climatização.
  - Segurança.
  - Comunicação.



# CGI (*Common Gateway Interface*)

---

```
int main(int argc, char *argv[]) {
    char value[128];

    printf("Content-type: text/html\n\n");
    printf("<h1>Red led remote control</h1>");

    if (read_parameter("cmd",value,10)) {
        if (strstr("ledon",value)) gpioclearbits(PORTA, PA3);
        if (strstr("ledoff",value)) gpiosetbits(PORTA, PA3);
    }

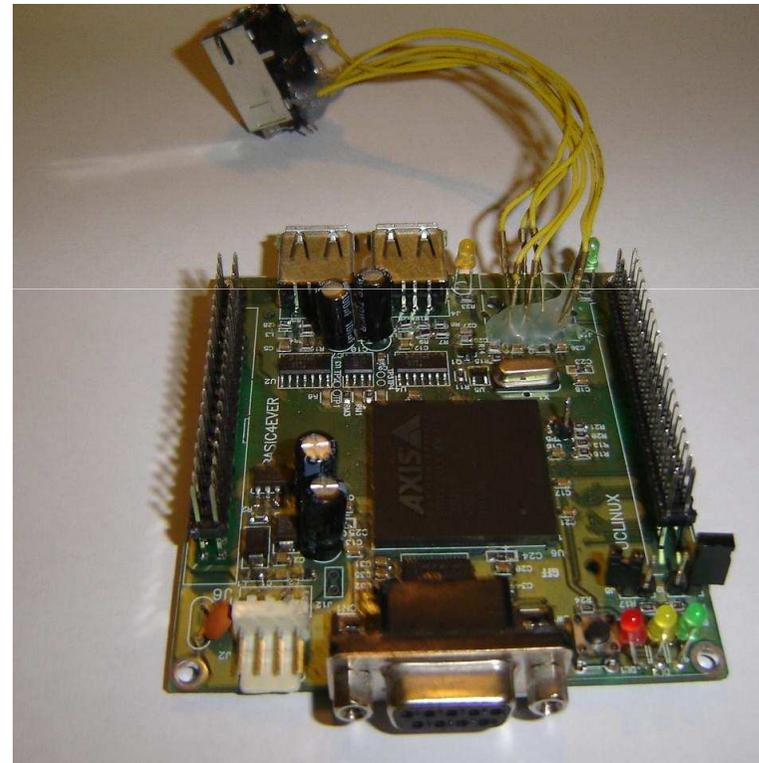
    printf("<a href=\"?cmd=ledon\">ON</a>");
    printf(" | ");
    printf("<a href=\"?cmd=ledoff\">OFF</a>");

    return 0;
}
```

# Fox Board

---

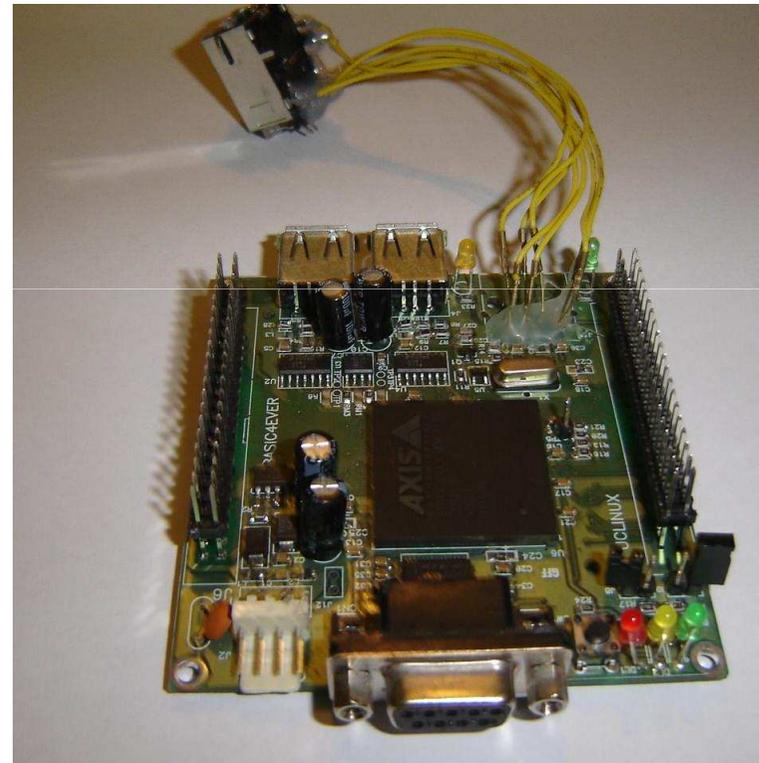
- Características de software:
  - Núcleo – Linux versão 2.4.31 (Padrão) ou 2.6.15.
  - Servidores - HTTP (WEB), FTP, SSH, TELNET.
  - Linguagens - C, C++, PHP, PYTHON, etc.
  - Compilador GNU C compiler, serviço disponível no site da ACME Systems



# Fox Board

---

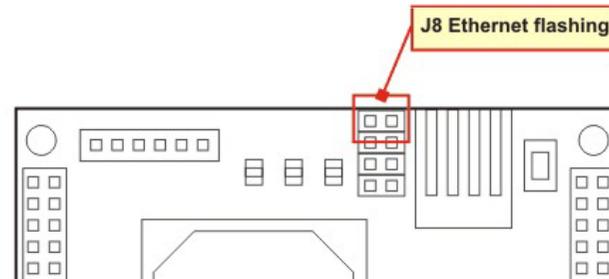
- Características de hardware:
  - CPU - Axis ETRAX 100LX 32 bit, RISC, 100MHz (100MIPS).
  - Memória - 4MB FLASH 16MB RAM.
  - Alimentação - 5 Volt 280mA (1watt)
  - Portas - 1 Ethernet (10/100 Mb/s)  
2 USB 1.1  
1 serial console port.
  - Extensões - 2 sockets IDE, SCSI, serial, paralela, linhas I/O, I2C bus interface.



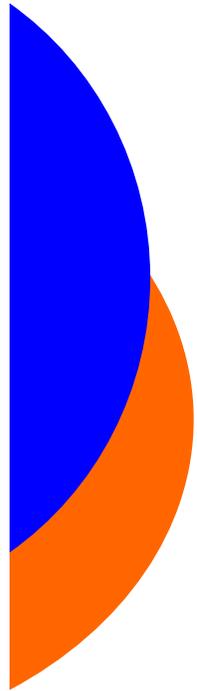
# Fox Board

---

- Instalação do Linux pela rede utilizando Windows XP:
  - Colocar *jumper* na J8 (Ethernet flashing).



- Requisitos:
  - WinPcap 4.0 instalado.
  - flashFOX.exe.

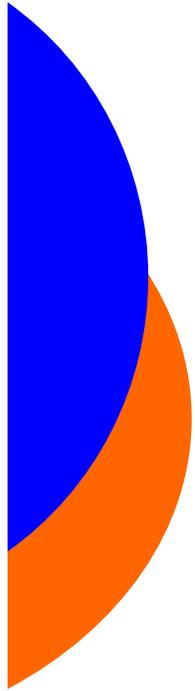


Vcc e Terra

Sensores

Tomadas,  
Iluminação  
Alarme

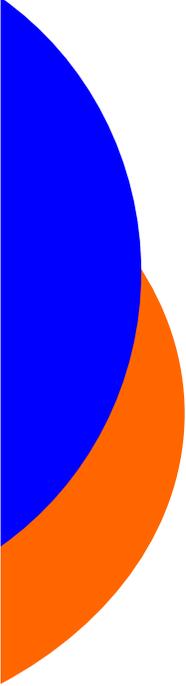
J7				J6				
Pin #	I/O Line	Default function		Def. Dir.	Pin #	I/O Line	Default function	Def. Dir.
1		GND	GND		1	+3.3V	Vcc (3.3V)	
2		GND	GND		2	+3.3V	Vcc (3.3V)	
3	IOG22	IOG22	Input/Output line G22	I	3	OG31	RTS3	Request to send /dev/ttyS3
4	IOG23	IOG23	Input/Output line G23	I	4	IG30	RXD3	Receive data /dev/ttyS3
5	IOG20	IOG20	Input/Output line G20	I	5	OG30	TXD3	Transmit data /dev/ttyS3
6	IOG21	IOG21	Input/Output line G21	I	6	IG31	CTS3	Clear to send/dev/ttyS3
7	IOG18	IOG18	Input/Output line G18	I	7	OG7	RTS2	Request to send /dev/ttyS2
8	IOG19	IOG19	Input/Output line G19	I	8	IG6	RXD2	Receive data /dev/ttyS2
9	IOG16	IOG16	Input/Output line G16	I	9	OG6	TXD2	Transmit data /dev/ttyS2
10	IOG17	IOG17	Input/Output line G17	I	10	IG7	CTS2	Clear to send/dev/ttyS2
11	OG26		VMO line on USB2		11		NMI	Non maskerable Interrupt
12	OG29	OG29	Output line G29	O	12		+5V	Vcc (5V)
13	OG25	OG25	Output line G25	O	13	IOG9	IOG9	Input/Output line G9
14	IG25		VM line on USB2		14	IOG8	IOG8	Input/Output line G8
15	OG28	OG28	Output line G28	O	15	IOG11	IOG11	Input/Output line G11
16	OG27		VPO line on USB2		16	IOG10	IOG10	Input/Output line G10
17	IG28		SPEED line on USB2		17	IOG13	IOG13	Input/Output line G13
18	IG27		RCV line on USB2		18	IOG12	IOG12	Input/Output line G12
19	IG26		VP line on USB2		19	IOG15	IOG15	Input/Output line G15
20	IG29		OE line on USB2		20	IOG14	IOG14	Input/Output line G14
21	IOG24	IOG24	Input/Output line G24	I	21	OG2	OG2	Output line G2 (I2C_RESET)(1)
22	IOG0	IOG0	Input/Output line G0 (I2C_RES)(1)	O	22	OG5	OG5	Output line G5
23			SPEED line on USB1		23	OG1	OG1	Output line G1
24			RCV line on USB1		24	IG1	IG1	Input line G1
25			OE line on USB1		25	OG4	OG4	Output line G4
26			VP line on USB1		26	OG3	OG3	Output line G3
27	RESET		Reset line (Out)	O	27	IG4	IG4	Input line G4
28	INTA		INTA (Out)	O	28	IG3	IG3	Input line G3
29	+5V		Vcc (5V)		29	IG2	IG2	Input line G2
30	IRQ		Interrupt request line	I	30	IG5	IG5	Input line G5
31	PA7	DCD2	Data carrier detect /dev/ttyS2	I	31	PB1	I2C_CLK	IC2 Clock
32	PA6	DSR2	Data set ready /dev/ttyS2	I	32	PB0	I2C_DATA	IC2 Data
33	PA5	RI2	Ring indicator /dev/ttyS2	I	33	PB3		VMO line on USB1
34	PA4	DTR2	Data terminal ready /dev/ttyS2	O	34	PB2		VPO line on USB1
35	PA3	DL1	Connected to DL1 Red led	O	35	PB5		VM line on USB1
36	PA2	DL2	Connected to DL2 Yellow led	O	36	PB4	PB4	Input/Output line B4
37	PA1	SW1	Connected to SW1	I	37	PB7	PB7	Input/Output line B7
38	PA0	PA0	Input/Output line A0	I	38	PB6	PB6	Input/Output line B6
39	+3.3V		Vcc (3.3V)		39		GND	GND
40	+3.3V		Vcc (3.3V)		40		GND	GND



## Trabalhos correlatos

---

- Sistema para automação e controle residencial via e-mail, Censi (2001).
- Protótipo de sistema de monitoramento remoto utilizando TCP/IP sobre Ethernet (802.3), Montibeller Junior (2005).
- Robotbuilder, Mammana e Cardoche (2007).

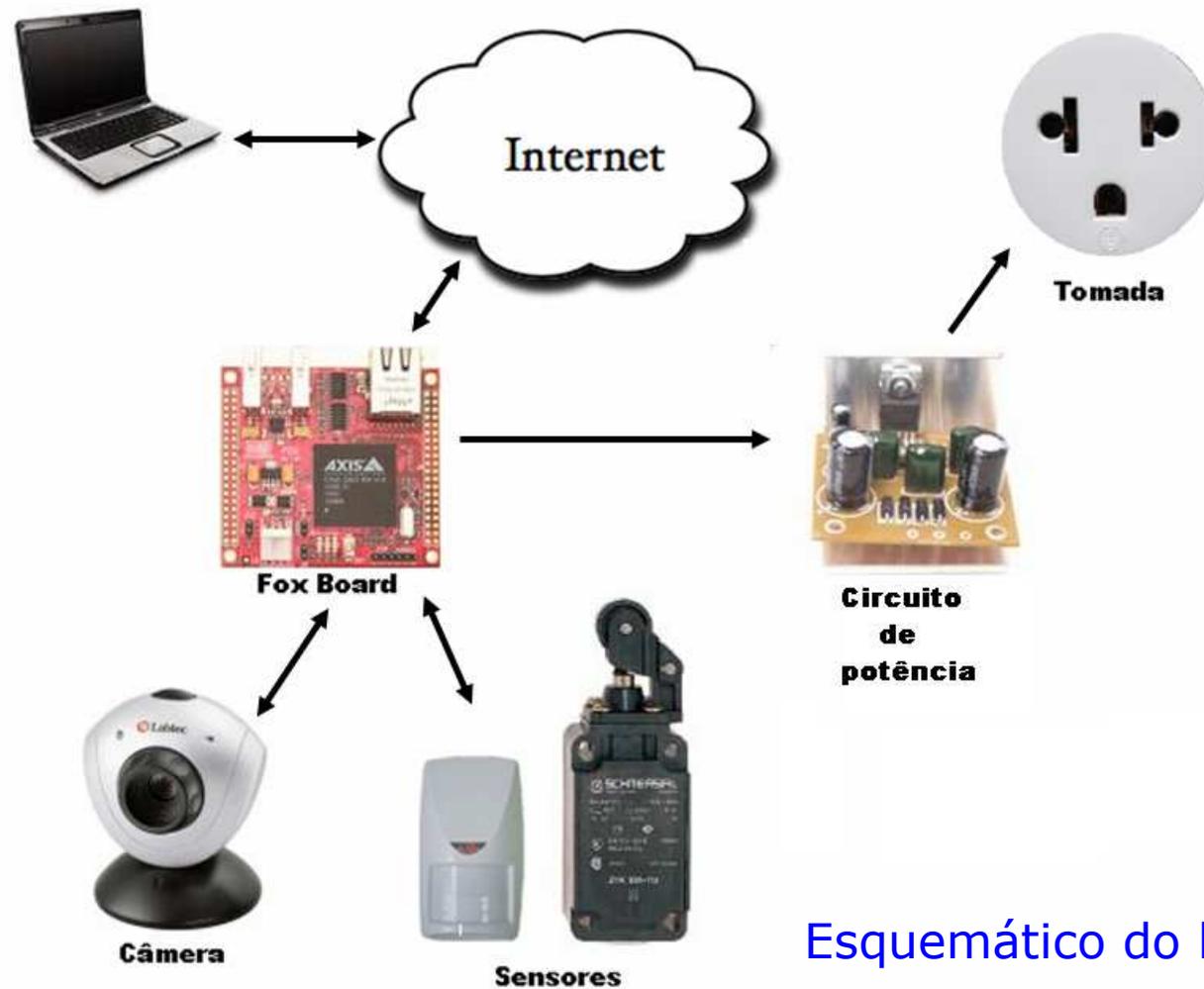


# Desenvolvimento do trabalho

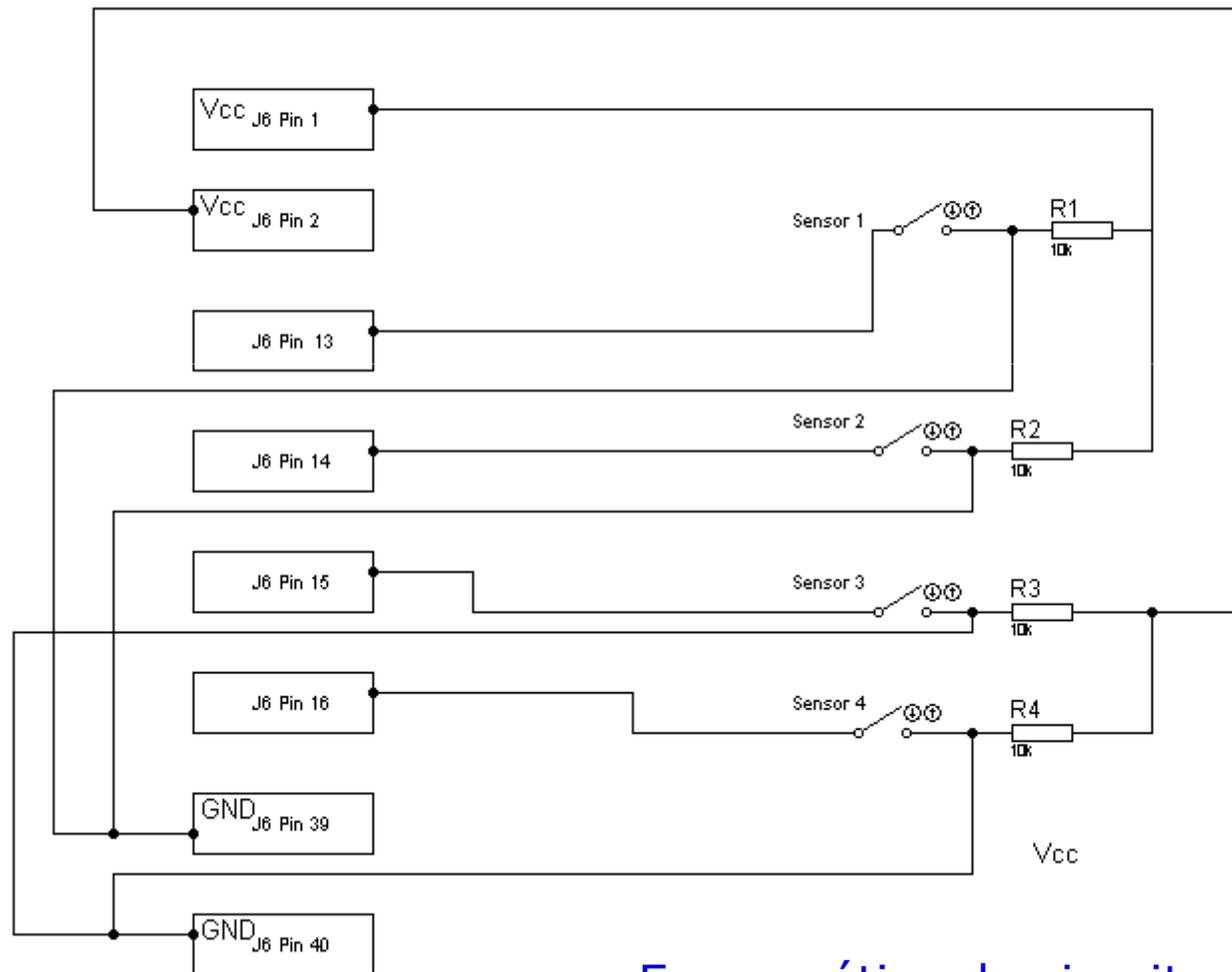
---

- Requisitos funcionais (RF) e os requisitos não funcionais (RNF).
  - monitorar os estados dos sensores (RF);
  - executar o servidor web (RF);
  - utilizar o sistema operacional Linux (RNF);
  - possibilitar a visualização das imagens captadas pela câmera e sinais dos sensores (RF);
  - receber os dados enviados pelo cliente através da rede (RF);
  - enviar dados ao cliente através da rede (RF);
  - gerenciar o dispositivo (RF);
  - criação de CGI utilizando a linguagem de programação C e código HTML (RNF);
  - a aplicação web do cliente deve ser independente de sistema operacional (RNF).

# Especificação do hardware

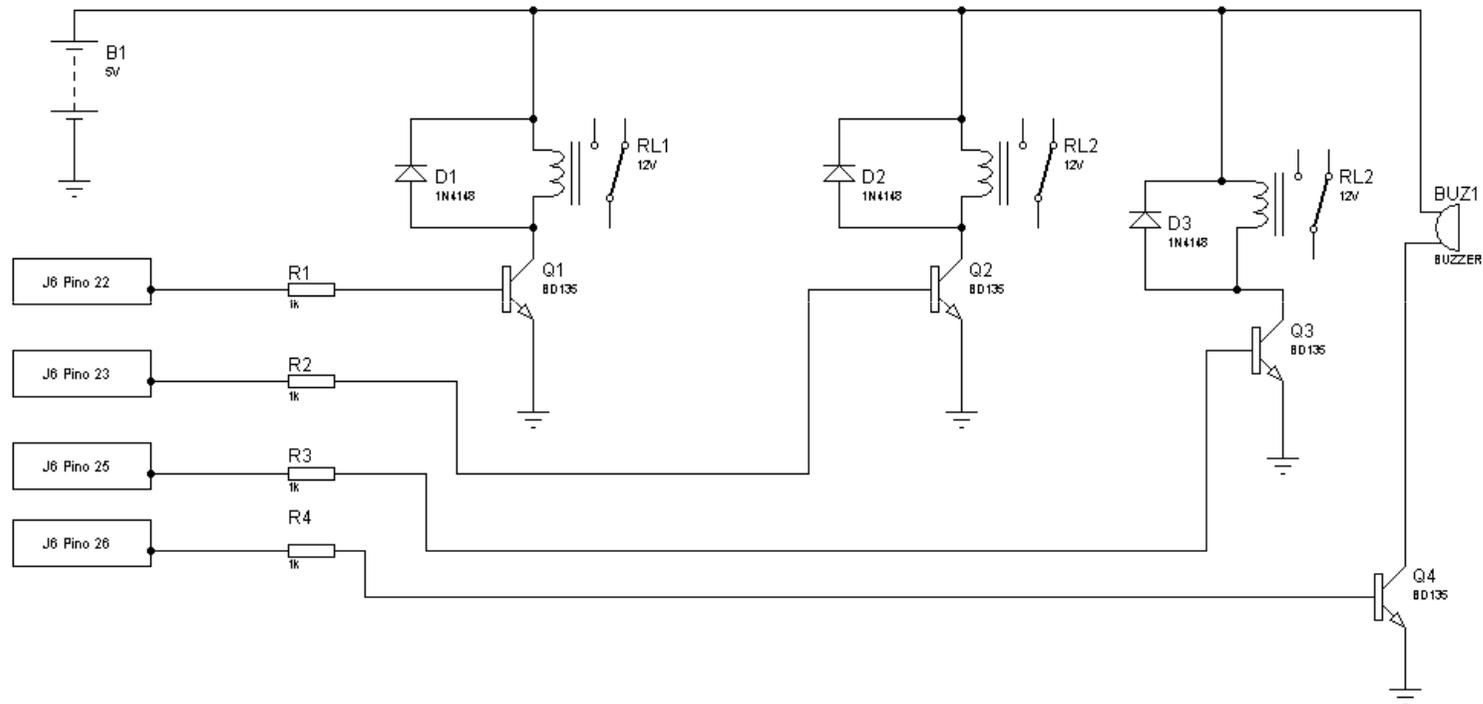


# Especificação do hardware



Esquemático do circuito de sensores

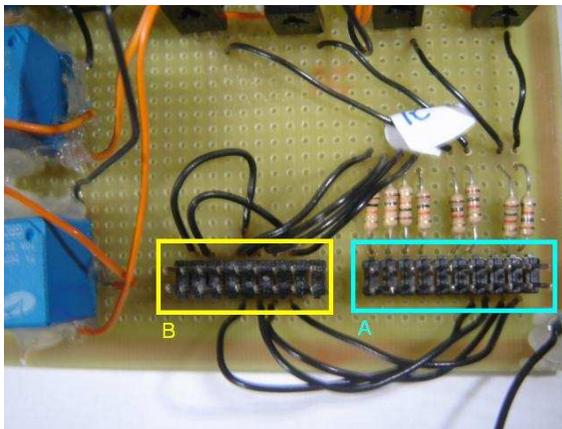
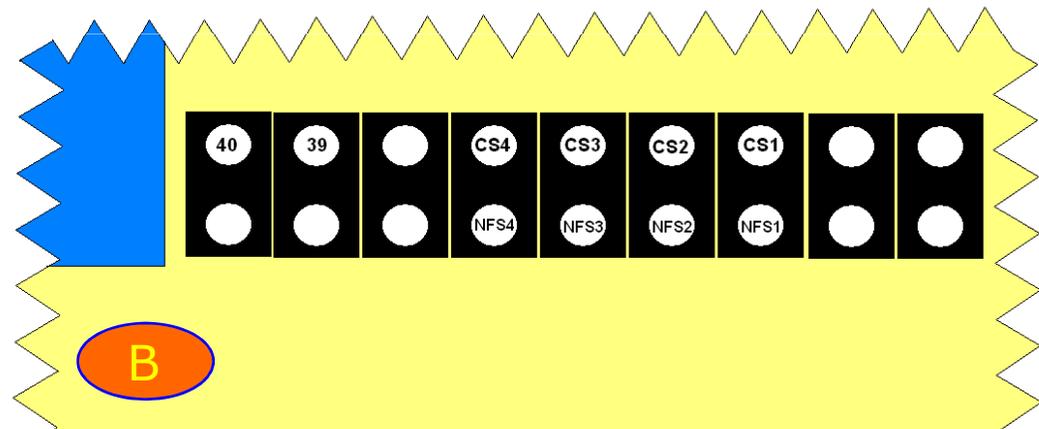
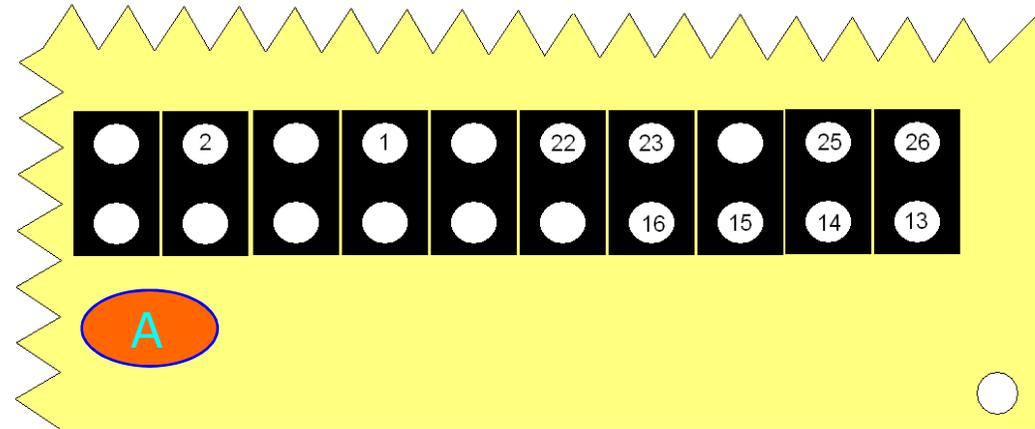
# Especificação do hardware

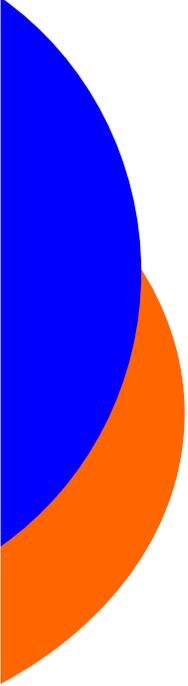


Esquemático do circuito tomadas e alarme

# Especificação do hardware

Pinos	Descrição
1,2,13,14,15,16,22,23,25,26,39,40.	Cada pino deve ser ligado ao número correspondente na Fox Board.
CSX	Pinos comuns dos sensores.
NFSX	Pinos normalmente fechado dos sensores

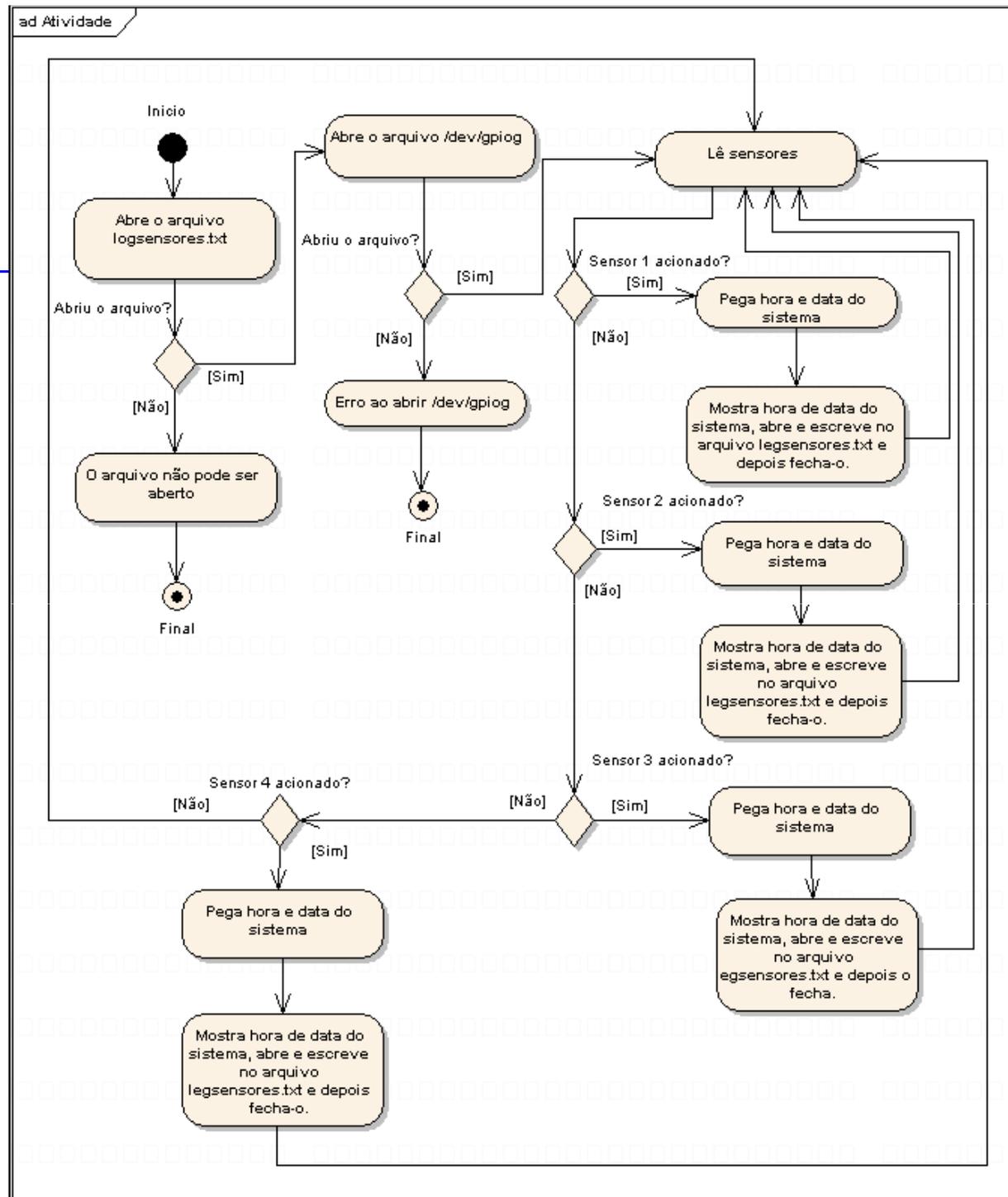
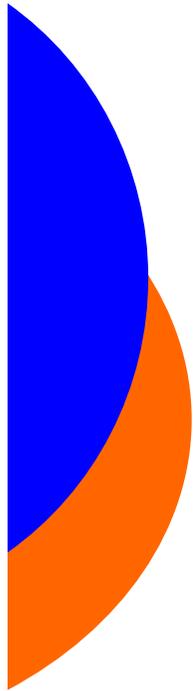


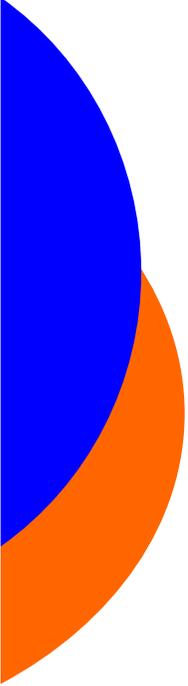


# Especificação do Software

---

- Diagrama de atividades da programação do leitor de sensores.

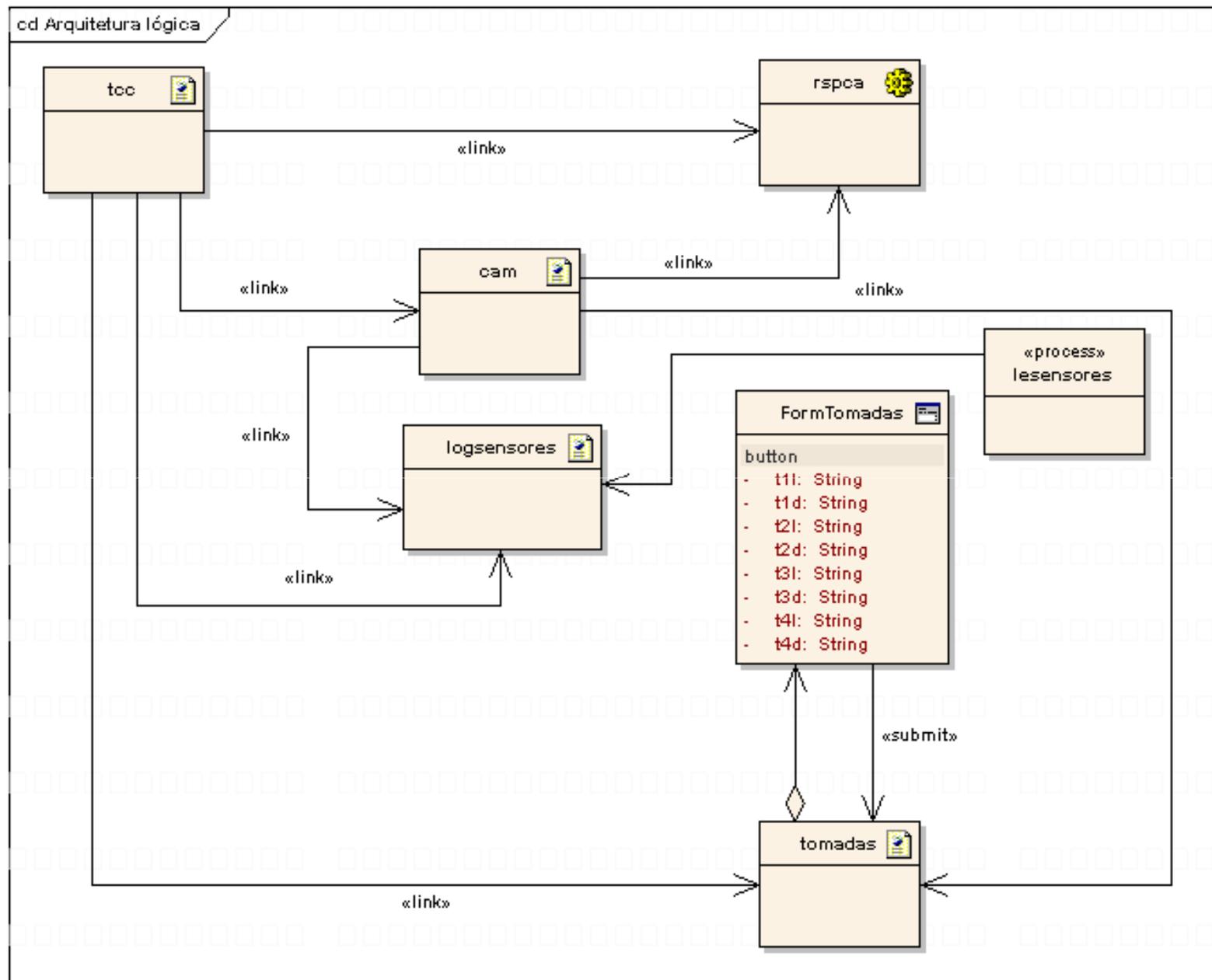
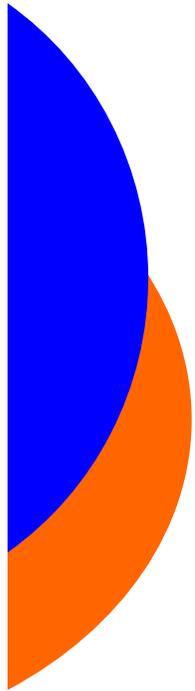


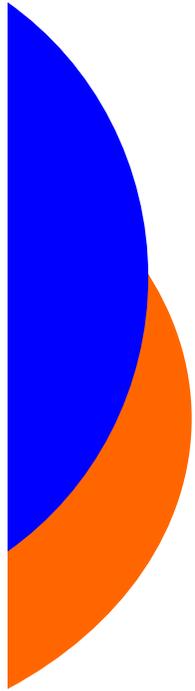


# Especificação do Software

---

- Diagrama de arquitetura lógica do sistema.



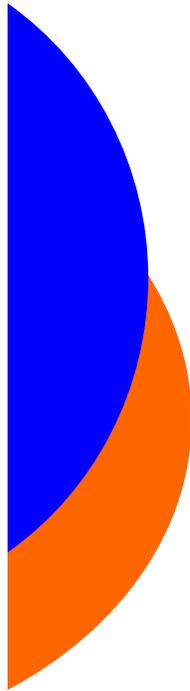


## Técnicas e ferramentas utilizadas

---

- flash FOX.
- WinPcap.
- Linguagem C.
- CGI.
- HTML.
- Web Compiler da Acme Systems.
- Enterprise Architect.
- Proteus Professional.





```
D:\ACME\WORK>flashFOX -d 2 -F -i my_fimage
```

```

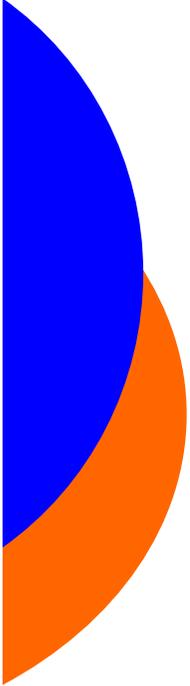
  _____
 /  _  _  _  \  /  _  _  _  \  /  _  _  _  \  /  _  _  _  \  /  _  _  _  \
|  (  (  (  (  |  (  (  (  (  |  (  (  (  (  |  (  (  (  (  |  (  (  (  (  |
 \  _  _  _  /  \  _  _  _  /  \  _  _  _  /  \  _  _  _  /  \  _  _  _  /
  _____


```

```
*****
e100boot made by axis.se, ported for windows by acmesystems.it
                        feedback, bugs : info@acmesystems.it
*****
```

```
Using device Broadcom NetXtreme Gigabit Ethernet Driver
Using internal boot loader: INTERNAL_NW - Network boot (default).
Starting boot...
We're doing a flash write, this may take up to a few minutes...
```

```
Device ID = 0x00001b49
This bootloader was built by blogic on Mon Mar 5 10:29:02 CET 2007.
Checksum of bootloader is 0x000a0ac0
Waiting for load info.
Checksum of file is 0x00001ebd
Got load info.
SET_REGISTER
0xb0000000
0x000095f8
SET_REGISTER
0xb0000004
0x00000004
SET_REGISTER
0xb000000c
0x09603737
SET_REGISTER
0xb0000008
...
```

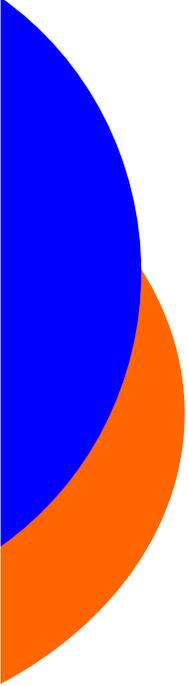


# Técnicas e ferramentas utilizadas: Servidor web BOA

---

```
1 WorkingRoot /
2 Port 80
3 User root
4 Group root
5 ErrorLog /dev/null
6 AccessLog /dev/null
7 UseLocaltime
8 DocumentRoot /usr/html
9 UserDir public_html
10 DirectoryIndex index.html
11 MimeTypes /etc/httpd/conf/mime.types
12 DefaultType text/plain
13 CGIPath /bin:/usr/bin
14 ScriptAlias /axis-cgi/ /usr/html/axis-cgi/
15 ScriptAlias /admin-bin/ /usr/html/admin-bin/
16 PasswdFile /etc/passwd
17 GroupFile /etc/group
18 Realm Administrator axadmin
19 ProtSpace Administrator /
20 ProtRealm Administrator anonymous
21 ProtPath Administrator /axis-cgi/admin/
22 ProtPath Administrator /usr/html/axis-cgi/admin/
23 ProtPath Administrator /admin-bin/
24 ProtPath Administrator /usr/html/admin-bin/
25 TransferCgi /axis-cgi/admin/flash
26 ScriptAlias /cgi/ /etc/httpd/cgi/
```

Arquivo: boa.conf



## Técnicas e ferramentas utilizadas: servidor web BOA

---

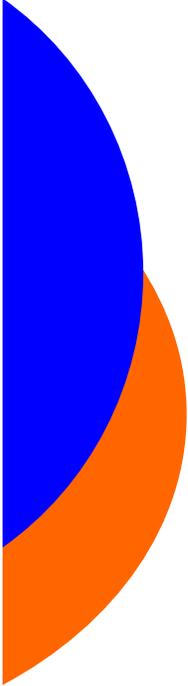
- Destacando as linhas:

- 8: `DocumentRoot /usr/html`

O parâmetro *DocumentRoot* define a pasta pública usada pelo servidor web.

- 26: `ScriptAlias /cgi/ /etc/httpd/cgi/`

O parâmetro *ScriptAlias* define a pasta que estarão os scripts CGI.



# Técnicas e ferramentas utilizadas:

---

- Compilador web da Acme Systems

**Web Compiler**  
*Simple web front end for Cris GCC compiler Based on FOX SDK 1.0 release 167*

1. Select the C source file from your hard disk...
2. Select the library linking mode:  dynamic  static
3. Do you need pthread lib ?  No  Yes
4. Do you need math lib ?  No  Yes
5. Press  to compile it and get the executable file.

---

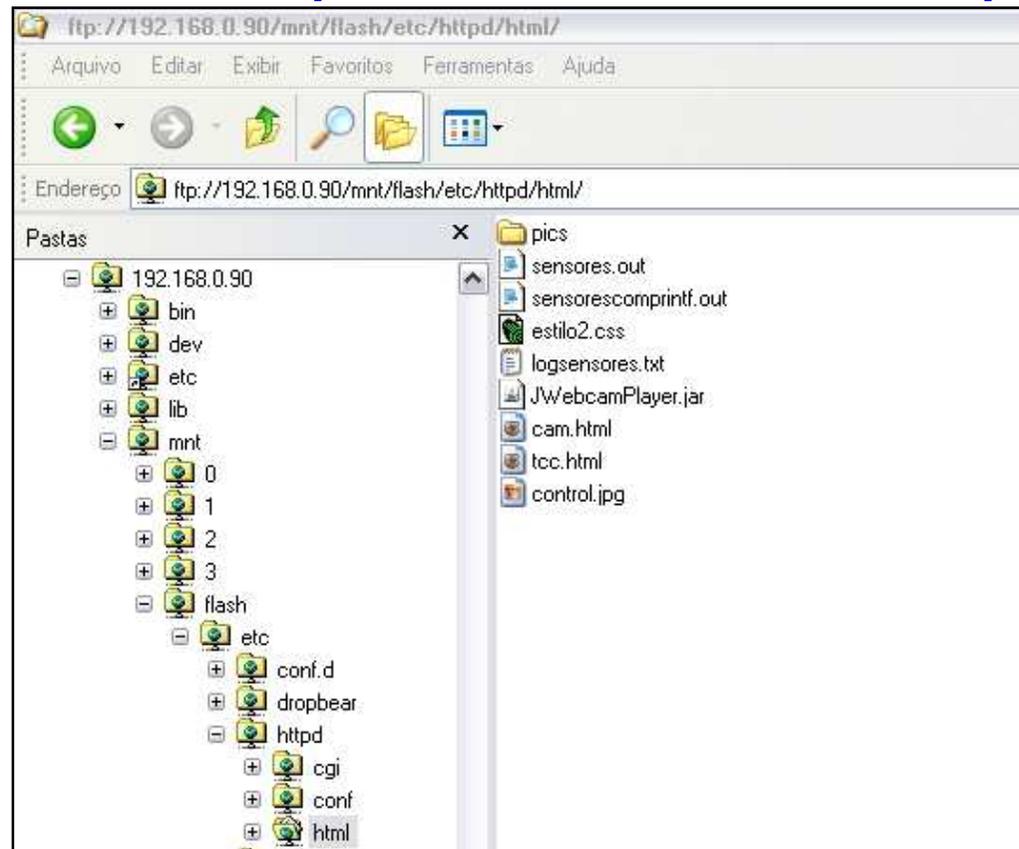
**Related links**

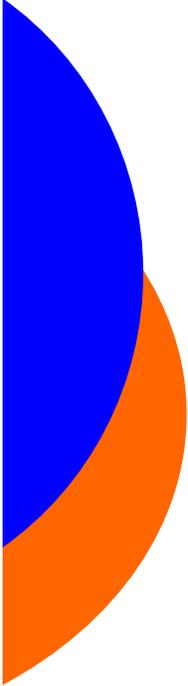
- [How to compile a C program with WebCompiler](#)

# Técnicas e ferramentas utilizadas:

---

- Arquivos compilados enviados por FTP.



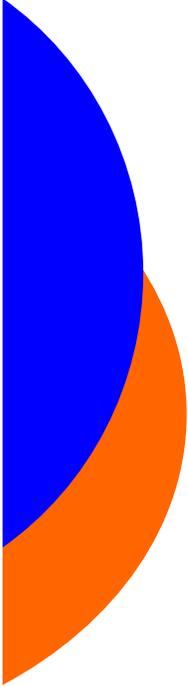


## Partes da implementação (Sensores)

---

```
if ((fd = open("/dev/gpiog", O_RDWR)) < 0) {  
    //printf("Erro ao abrir /dev/gpiog\n");  
    exit(0);  
}
```

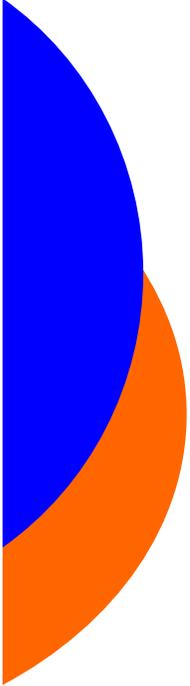
```
//Seta a linha 9 da PortG pin # 13  
iomask1=1<<9;  
//Seta a linha 8 da PortG pin # 14  
iomask2=1<<8;  
//Seta a linha 11 da PortG pin # 15  
iomask3=1<<11;  
//Seta a linha 9 da PortG pin # 16  
iomask4=1<<10;
```



# Partes da implementação (Sensores)

---

```
while (1) {
    //lendo a entrada
    value=ioctl(fd, _IO(ETRAXGPIO_IOCTLTYPE, IO_READBITS));
    //se o sensor 1 for acionado
    lt = time (NULL);
    if ((value&iomask1)==0) {
        //Abre o arquivo e adiciona novo conteúdo, caso o arquivo ainda não exista ele é criado
        abrearquivo();
        //pega a data e hora do sistema
        ptr = localtime(&lt);
        //aux recebe a data e hora
        sprintf(aux, asctime(ptr));
        //printf("Janela aberta em: ");
        //printf(aux);
        //escreve o conteúdo da variavel source no arquivo
        fputs("Janela aberta em: ", source);
        fputs(aux, source);
        //fecha arquivo
        fclose(source);
    }
    if ((value&iomask2)==0) {
```

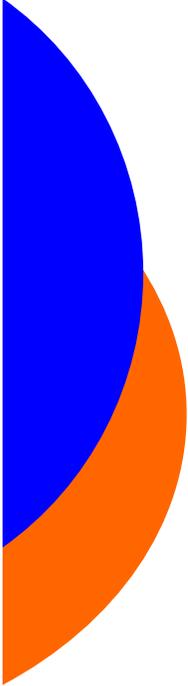


# Partes da implementação (Tomadas)

---

```
printf("Content-type: text/html\n\n");
printf("<html>\n");
printf("<head>\n");
printf("<title>\n");
printf("TCC - Daniel Baumann");
printf("</title>\n");
printf("</head>\n");
printf("<body>\n");
if (query_string!=NULL) {
    commandExecute(query_string);
}

printf("<form>\n");
printf("<TABLE WIDTH='0%' BORDER='0'>");
printf("<TR><TD>");
printf("<INPUT TYPE='submit' NAME='Submit' VALUE='Tomada 1 Ligar'></TD><TD>");
printf("<INPUT TYPE='submit' NAME='Submit2' VALUE='Tomada 1 Desligar'></TD>");
printf("</TR>");
printf("<TR><TD>");
```



## Partes da implementação (Tomadas)

---

```
void commandExecute(char *query_string) {
    char *p;
    //Tomada 1
    if ((p=strstr(query_string,"Tomada+1+Ligar")) {
        //printf("Liga tomada 1\n");
        int iomask;
        iomask=1<<5;
        ioctl(fd,_IO(ETRAXGPIO_IOCTLTYPE,IO_SETBITS),iomask);
        return;
    }
    if ((p=strstr(query_string,"Tomada+1+Desligar")) {
        //printf("Desligar tomada 1\n");
        int iomask;
        iomask=1<<5;
        ioctl(fd,_IO(ETRAXGPIO_IOCTLTYPE,IO_CLRBITS),iomask);
        return;
    }
}
```

# Estudo de caso – Detecção de porta aberta

The image shows two browser windows. The left window displays a web interface for a residential security system. The right window shows a log file with sensor events.

**Sistema De Segurança Residencial**

- Home
- Servfox
- >Ver Câmera
- Log sensores
- Tomadas

**Linux embarcado**

**Mozilla Firefox**

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

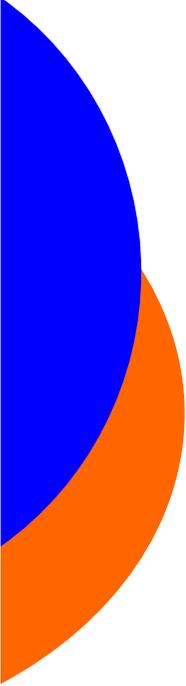
http://192.168.0.90/logsensores.txt

Movimento detectado em:	Thu Jul 3 17:00:43 2008
Movimento detectado em:	Thu Jul 3 17:00:44 2008
Movimento detectado em:	Thu Jul 3 17:01:51 2008
Movimento detectado em:	Thu Jul 3 17:01:52 2008
Porta aberta em:	Thu Jul 3 17:02:05 2008
Porta aberta em:	Thu Jul 3 17:02:06 2008
Porta aberta em:	Thu Jul 3 17:02:07 2008
Janela aberta em:	Thu Jul 3 17:02:11 2008
Movimento detectado em:	Thu Jul 3 17:02:15 2008
Movimento detectado em:	Thu Jul 3 17:02:16 2008
Movimento detectado em:	Thu Jul 3 17:02:17 2008
Movimento detectado em:	Thu Jul 3 17:02:18 2008
Movimento detectado em:	Thu Jul 3 17:02:24 2008
Movimento detectado em:	Thu Jul 3 17:02:26 2008
Movimento detectado em:	Thu Jul 3 17:02:27 2008
Movimento detectado em:	Thu Jul 3 17:02:29 2008
Movimento detectado em:	Thu Jul 3 17:02:34 2008
Movimento detectado em:	Thu Jul 3 17:02:35 2008
Movimento detectado em:	Thu Jul 3 17:02:39 2008
Movimento detectado em:	Thu Jul 3 17:02:40 2008
Movimento detectado em:	Thu Jul 3 17:02:45 2008
Movimento detectado em:	Thu Jul 3 17:02:47 2008
Movimento detectado em:	Thu Jul 3 17:02:48 2008
Movimento detectado em:	Thu Jul 3 17:02:49 2008
Movimento detectado em:	Thu Jul 3 17:02:51 2008
Movimento detectado em:	Thu Jul 3 17:02:52 2008
Movimento detectado em:	Thu Jul 3 17:02:53 2008
Movimento detectado em:	Thu Jul 3 17:02:54 2008
Movimento detectado em:	Thu Jul 3 17:02:55 2008
Movimento detectado em:	Thu Jul 3 17:02:57 2008
Movimento detectado em:	Thu Jan 1 00:08:14 1970
Movimento detectado em:	Thu Jan 1 00:08:15 1970
Movimento detectado em:	Fri Jul 4 12:20:10 2008
Movimento detectado em:	Fri Jul 4 12:20:12 2008
Movimento detectado em:	Fri Jul 4 12:20:23 2008
Movimento detectado em:	Fri Jul 4 12:20:24 2008
Movimento detectado em:	Fri Jul 4 12:20:27 2008
Movimento detectado em:	Fri Jul 4 12:20:28 2008
Movimento detectado em:	Fri Jul 4 12:20:29 2008
Porta aberta em:	Fri Jul 4 12:20:33 2008
Porta aberta em:	Fri Jul 4 12:20:34 2008

Concluído

# Estudo de caso – Iluminação

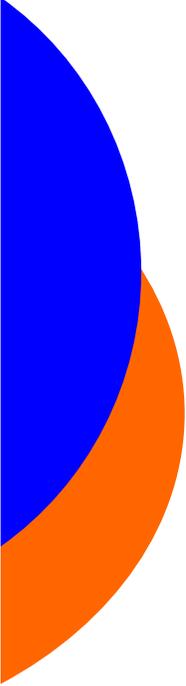
The image displays a web browser window titled "TCC - Daniel Baumann - Mozilla Firefox" with the address bar showing "http://192.168.0.90/#popup". The main content area is titled "Sistema De Segurança Residencial" and features a navigation menu on the left with options: Home, Servfox, Ver Câmera, Log sensores, and Tomadas. The central control panel includes buttons for "Tomada 1 Ligar", "Tomada 1 Desligar", "Tomada 2 Ligar", "Tomada 2 Desligar", "Acender luzes", "Apagar luzes", "Disparar alarme", and "Desligar alarme". Below these buttons is a status indicator showing ">>" and "Concluído". Two inset images are connected to the interface: a red-bordered image on the left shows a glowing light fixture, and a blue-bordered image on the right shows a camera view of a light fixture with wires. A blue bar at the bottom of the interface is labeled "Linux embarcado".



# Resultados e discussão

---

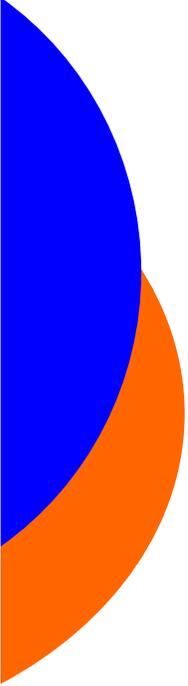
- **A câmera ligada a Fox Board apresentou um resultado satisfatório em relação o tempo de resposta.**
- **Apresentou bons resultados realizando corretamente a leitura dos sensores.**
- **O tempo de resposta do momento que é pressionado um botão no navegador para acionar ou desacionar tomadas, alarme e iluminação também apresentou um resultado muito satisfatório.**
- **Apesar da limitação de 4MB FLASH e 16MB RAM do modelo de Fox Board utilizado não foi encontrado problema algum de falta de espaço.**
- **Bom material de referência sobre a placa Fox Board.**
- **Ótimo suporte técnico da Acme Systems.**



# Conclusão

---

- O tempo gasto para encontrar a causa e soluções do problema e como conciliar todas as funcionalidades em uma só imagem de Linux foi muito grande, o que prejudicou muito o tempo de desenvolvimento do protótipo.
- Queima da primeira Fox Board também prejudicou muito o tempo de desenvolvimento do protótipo.
- Emprego de linux embarcado utilizando a placa Fox Board, o mercado ainda é novo no Brasil.
- O software do protótipo a nível de usuário é independente de plataforma já que é uma aplicação web.
- O navegador recomendado é o Mozilla Firefox 3 ou inferior.



# Extensões

---

- Utilizar o modelo de Fox Board LX832 com 8MB FLASH e 32MB RAM para poder fazer uso da ferramenta FoxServe (KDEV, 2007), que possui PHP 5.0.5 e SQLite.
- Utilizar applet que capture vídeo e/ou imagem e armazene.
- Instalar motor passo para movimentar a câmera.
- Substituir os cabos por rede sem fio.
- Detector de movimento pela câmera.

# PROTÓTIPO DE UM SISTEMA DE SEGURANÇA RESIDENCIAL COM LINUX EMBARCADO

---

