II SATEC

2ª Semana de
Atualização Técnica

Jun. 10-14, 2013
CASCAVEL-PR

Padrão de desenvolvimento para prototipagem rápida em projetos de sistemas de controle utilizando a plataforma de prototipagem eletrônica Arduino.

Engº Hamilton Sena

FIESC
CIESC
SESI
SENAI
IEL

SENAI

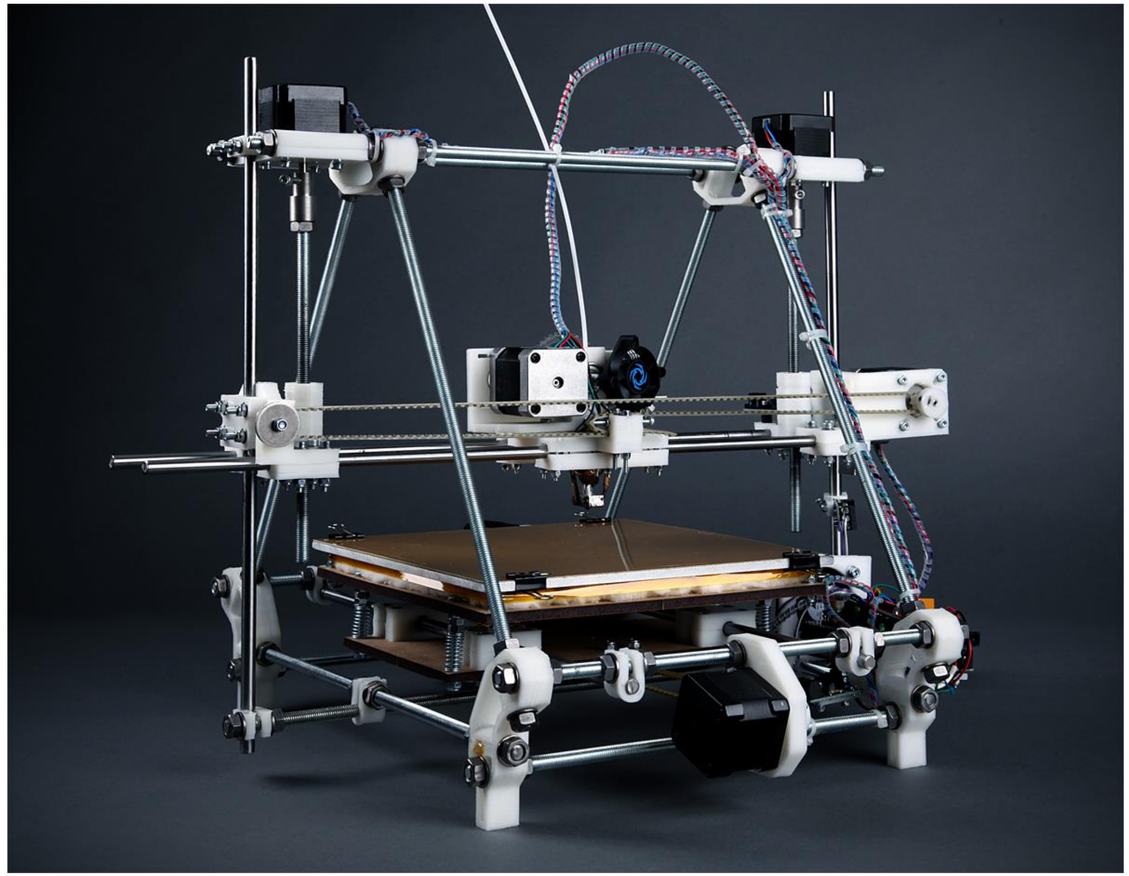
Hamilton Sena

- Acadêmica
  - Técnico em processamento de dados
  - Engenheiro de Controle e Automação
- Profissional
  - Técnico em informática
  - Técnico de reparo de equipamentos eletrônicos
  - Desenvolvedor e analista de sistemas
  - Desenvolvedor de sistemas embarcados
  - Sócio-proprietário da empresa Mobhis Automação Urbana Ltda.
  - Professor do Senai

MOBHIS
AUTOMAÇÃO URBANA

Prototipagem rápida

- **Prototipação** é uma abordagem baseada numa visão evolutiva do desenvolvimento.
- Com intuito de avaliar algumas de suas características antes que o sistema venha realmente a ser construído, de forma definitiva.



REPRAP - http://reprap.org

Prototipagem rápida - Vantagens

Time-to-market

Quem chega primeiro ganha o mercado

Ajuda a controlar o risco

É muito mais seguro decidir diante de um protótipo

Rápido e barato

Método mais rápido e barato de resolver problemas.

Projeto Toopedalando

MOBHIS
AUTOMAÇÃO URBANA

O mercado não espera!

Outubro/2010
Interesse

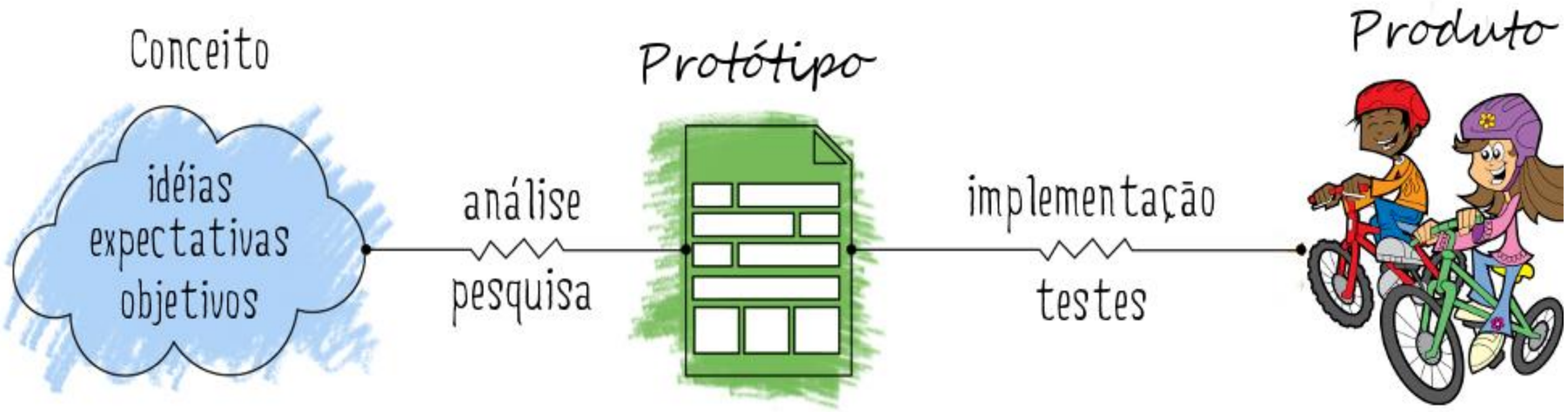
Novembro/2010
1ªapresentação

Março/2011
Lançamento do Edital

Julho/2011
Instalação 1ª Estação

1 mês

7 meses



Padrão de desenvolvimento

Ansioso

Mão na massa

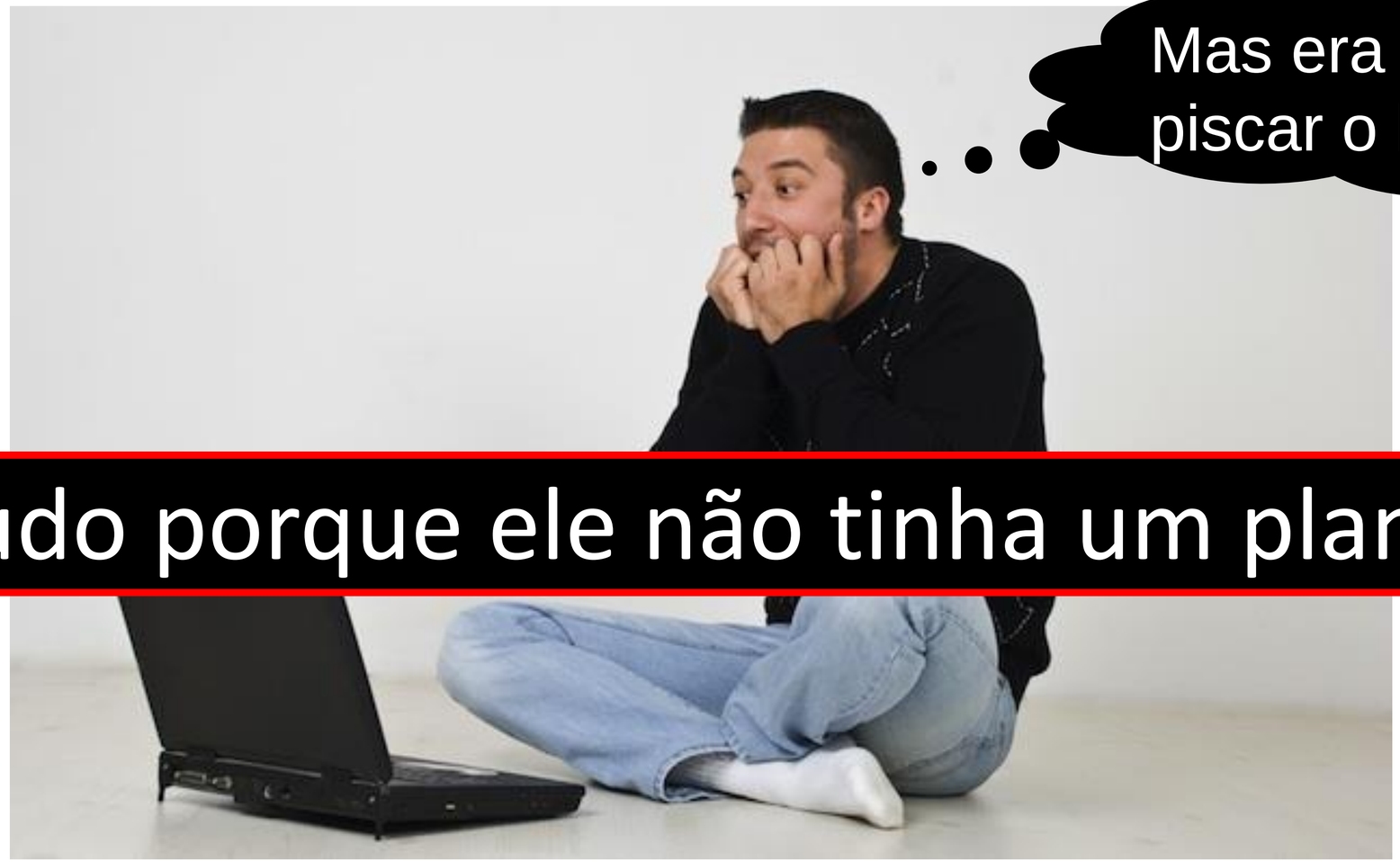
Depois de 2000 linhas ...

... começa a testar

O código nem compila

Caos total!

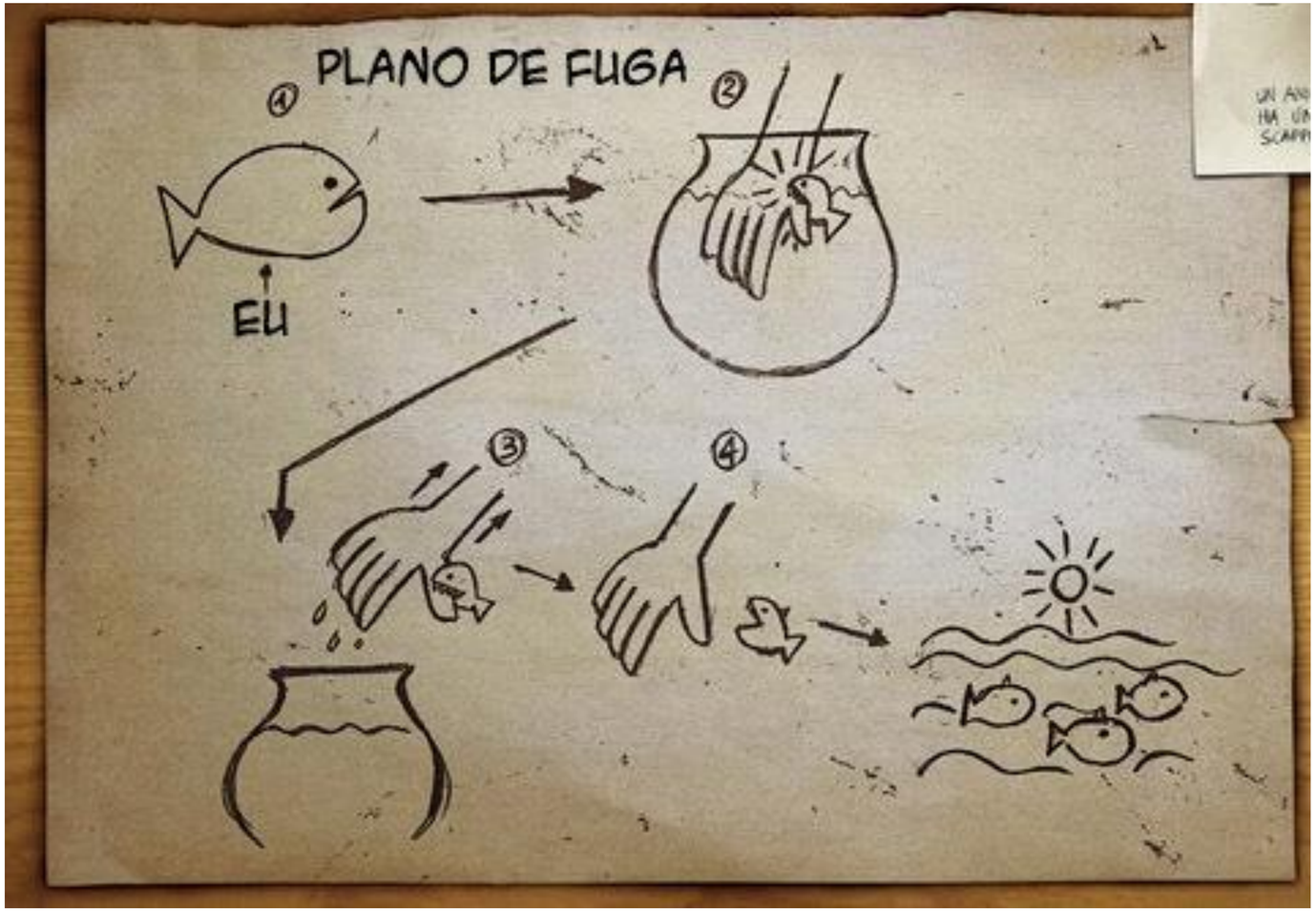
Padrão de desenvolvimento



Tudo porque ele não tinha um plano.

Padrão de desenvolvimento

Nunca comece nada sem um plano !!!

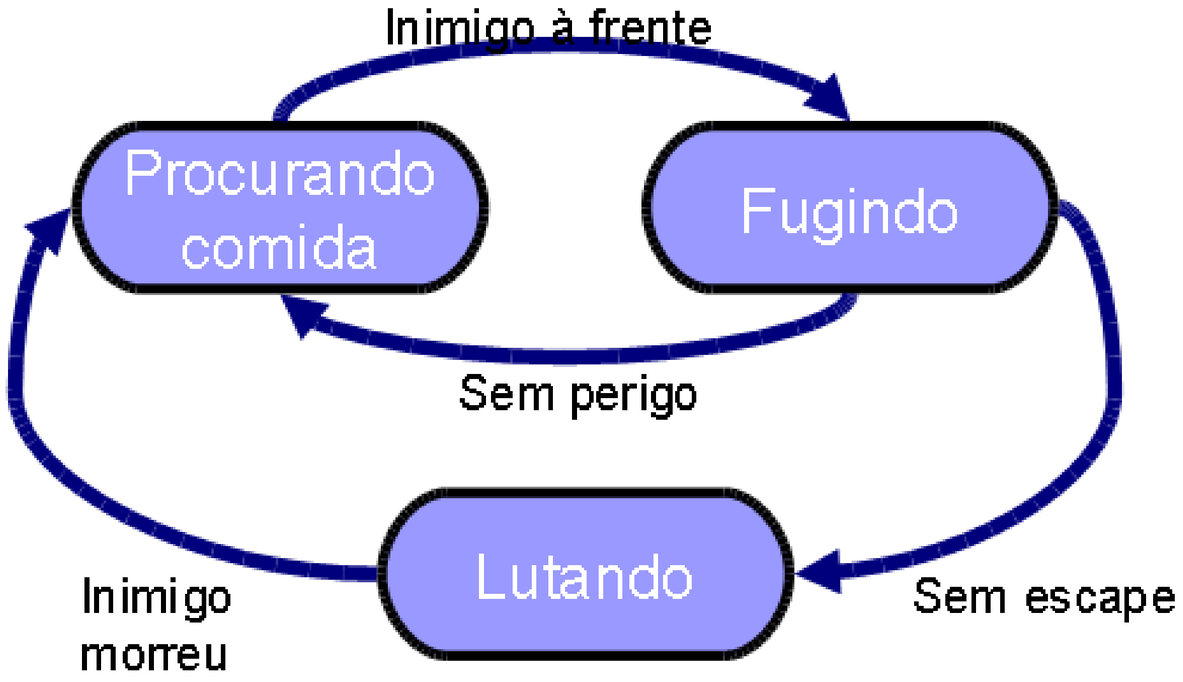


Padrão de desenvolvimento

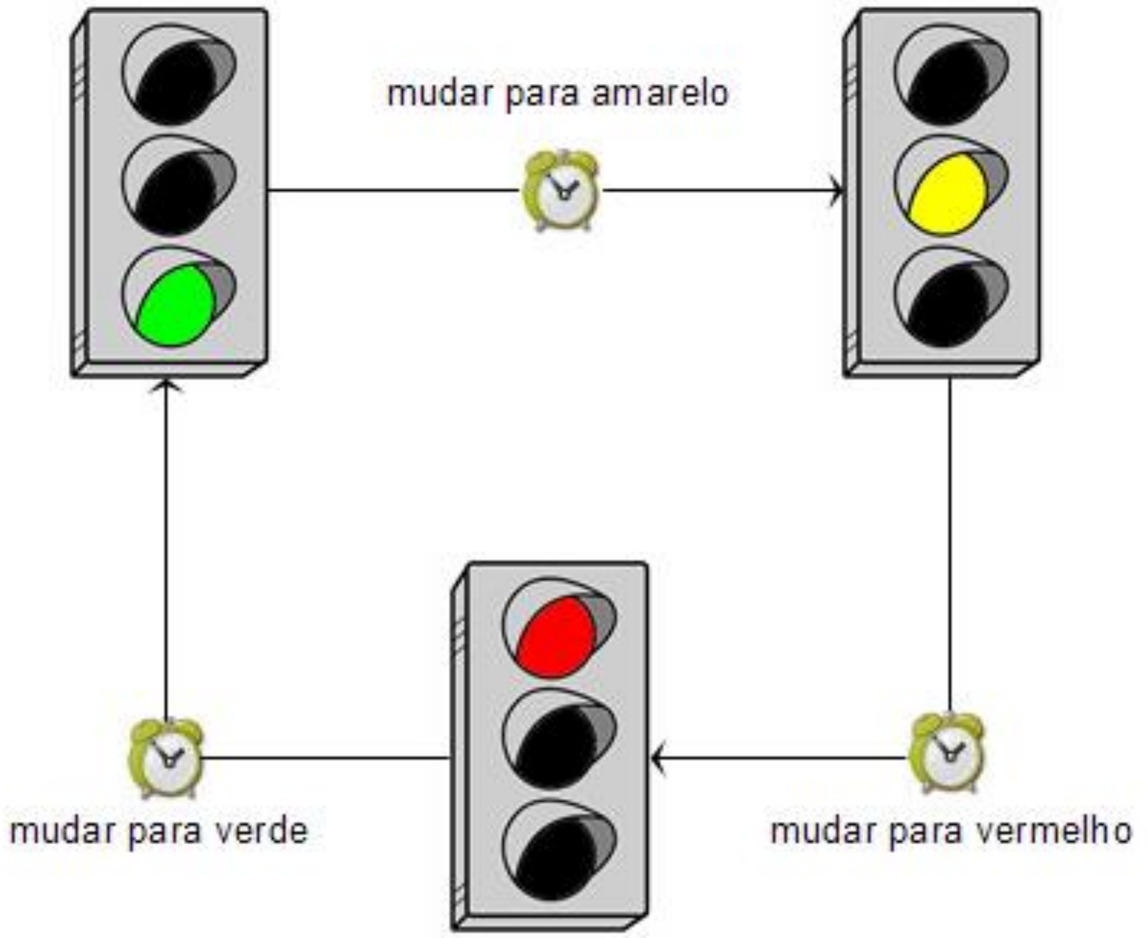
- São soluções para problemas normalmente encontrados em projetos de software.
- São independentes de linguagem.
- E oferecem uma descrição ou modelo de como resolver determinado problema.

Maquina de estados

São estruturas lógicas compostas por um conjunto de estados e um conjunto de regras de transição entre os estados.

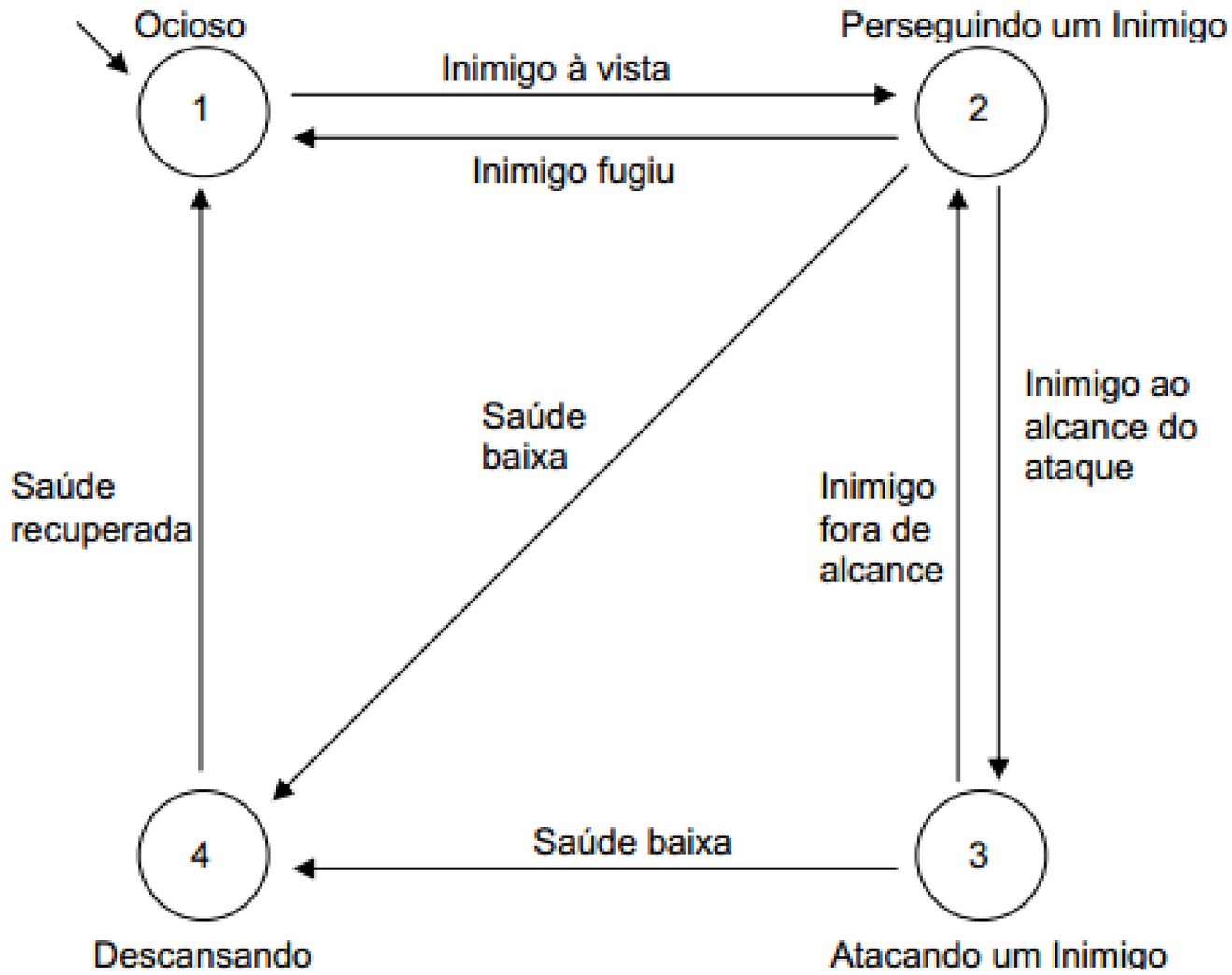


FSM – Controle de processos

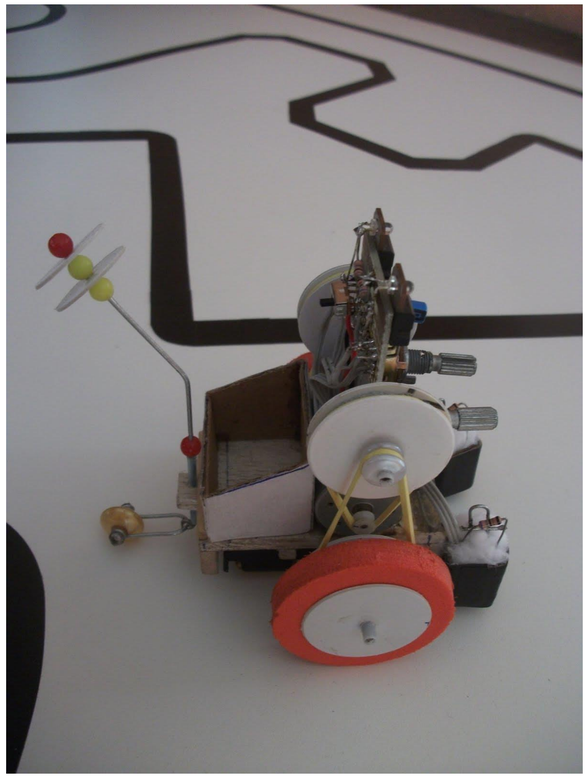


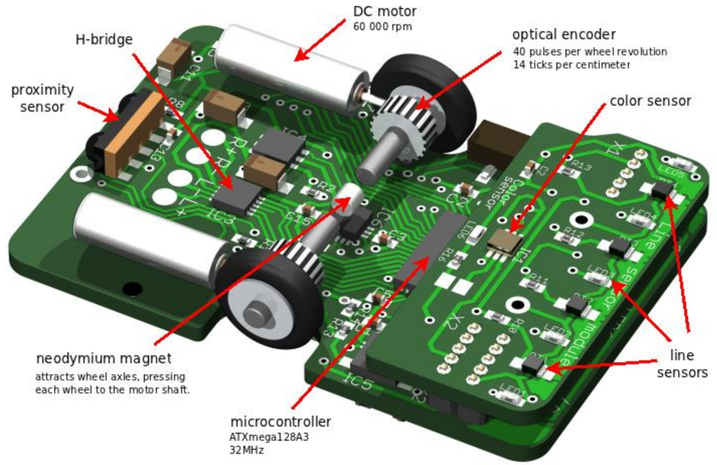
FSM - Jogos

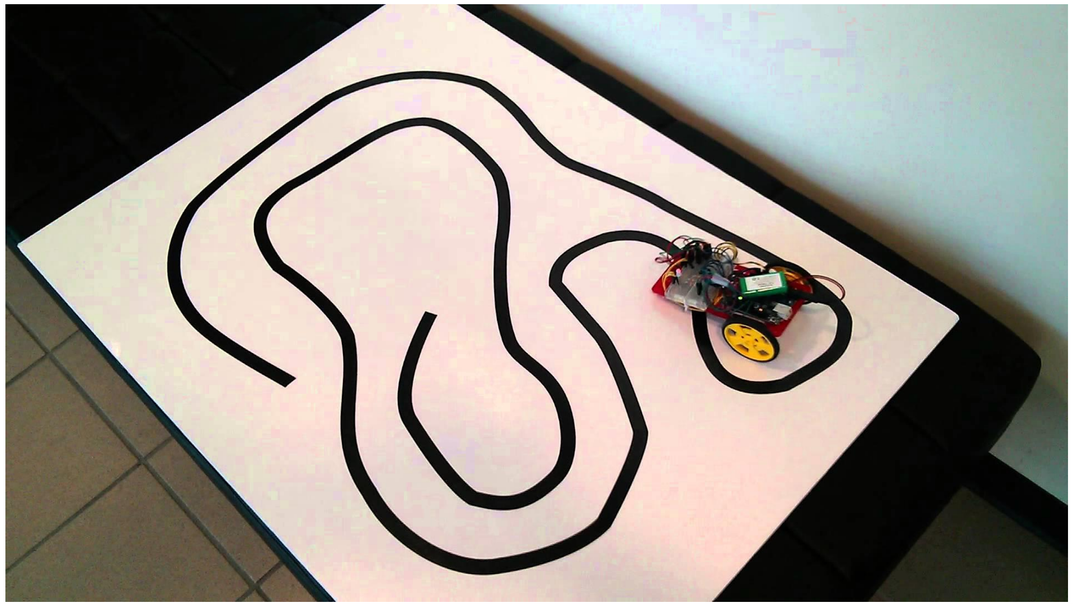

0
0
1:47
0
0


Ocioso
1
Perseguindo um Inimigo
2
Inimigo à vista
Inimigo fugiu
Saúde baixa
Saúde recuperada
Inimigo fora de alcance
Inimigo ao alcance do ataque
4
Descansando
Saúde baixa
3
Atacando um Inimigo

FSM – Robótica

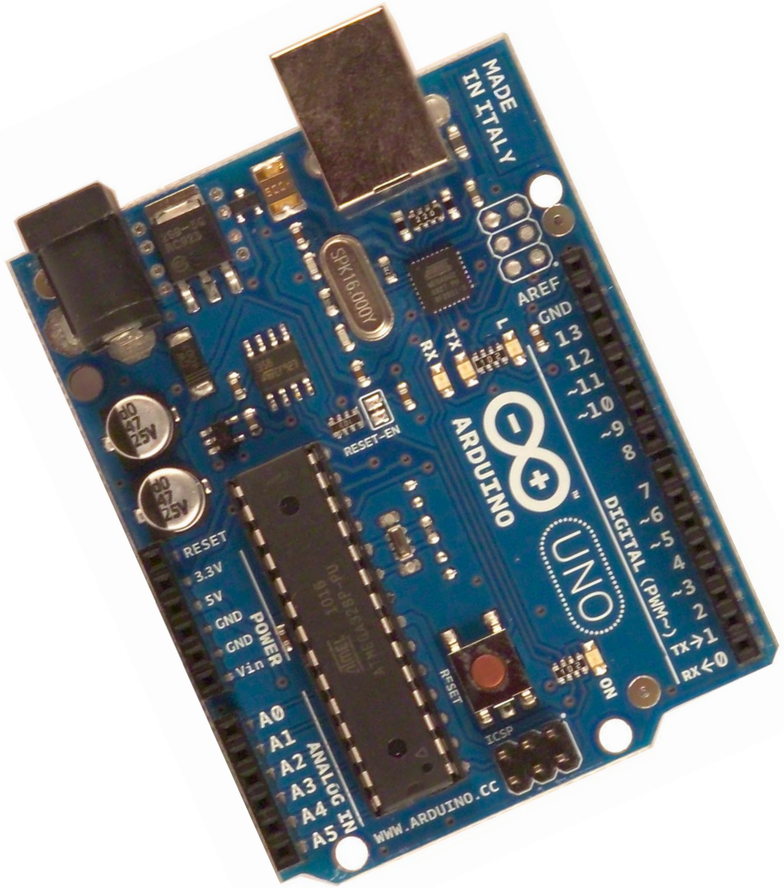




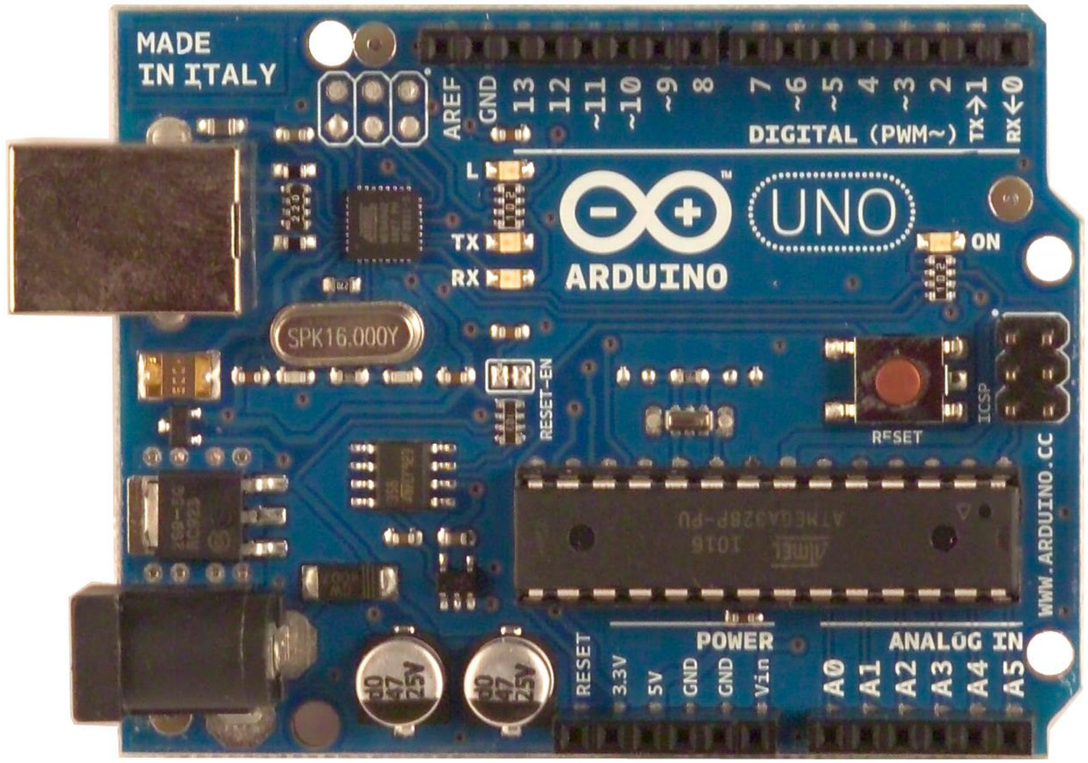


Arduino

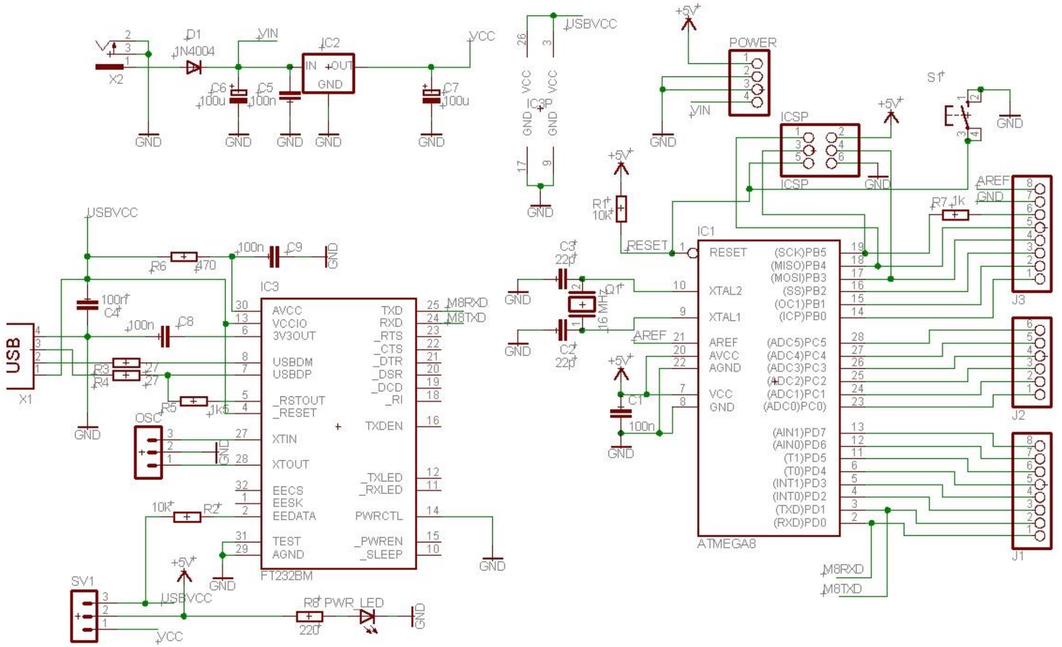
- Em 2005, um professor italiano, Mossimo Banzi, decidiu que queria um simples controlador lógico acessível aos seus alunos de forma a desenvolverem os seus próprios projetos técnicos.
- Consiste num microcontrolador Atmel AVR de 8 bits, com componentes complementares para facilitar a programação e incorporação para outros circuitos.
- E/S digital e analógica, além de uma interface USB, para interagir com computador, e programação.



Arduino



- **Hardware Aberto**



open hardware

Arduino

open hardware

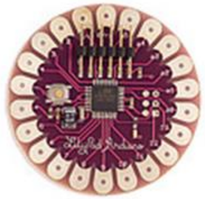
Arduino Uno

Arduino Leonardo

Arduino Ethernet

Arduino Pro

Arduino Mega 2560



Arduino LilyPad

Arduino BT

Arduino Nano

Arduino Mega ADK

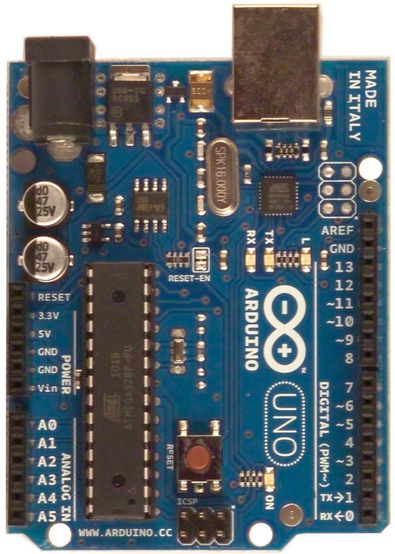
Arduino Fio

USB/Serial Light Adapter

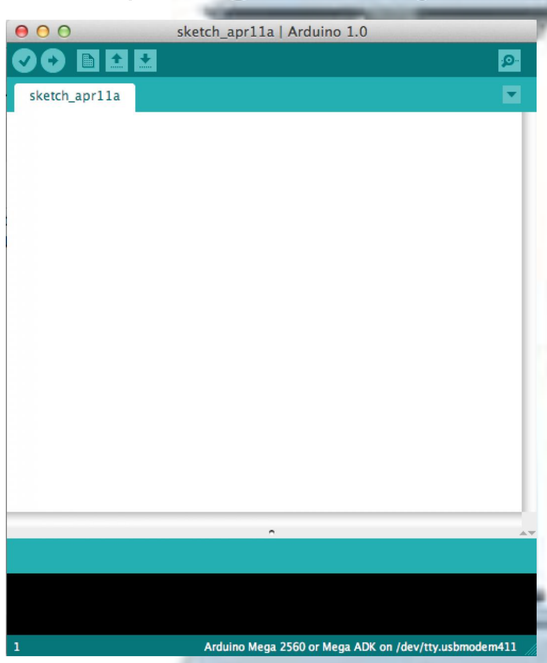
Arduino Mini

Universo Arduino

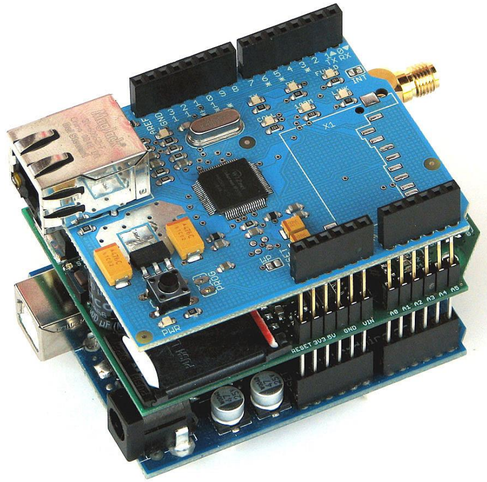
O Hardware



O ambiente de programação



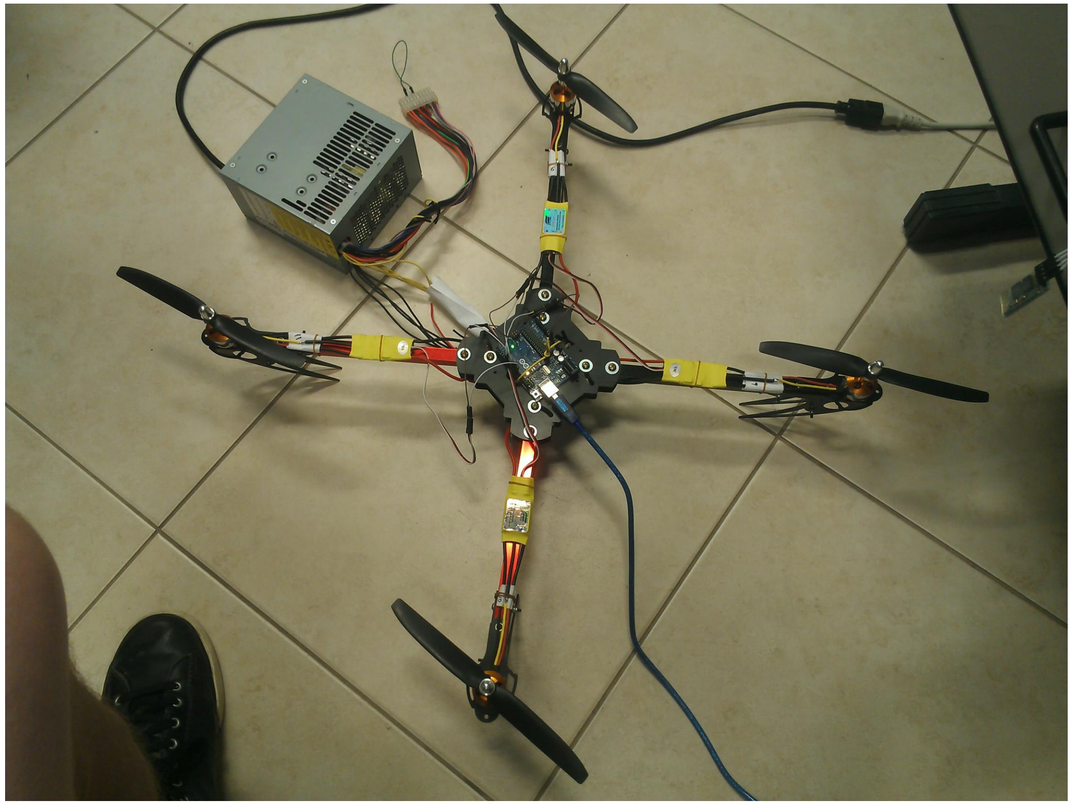
Os Shields



A comunidade

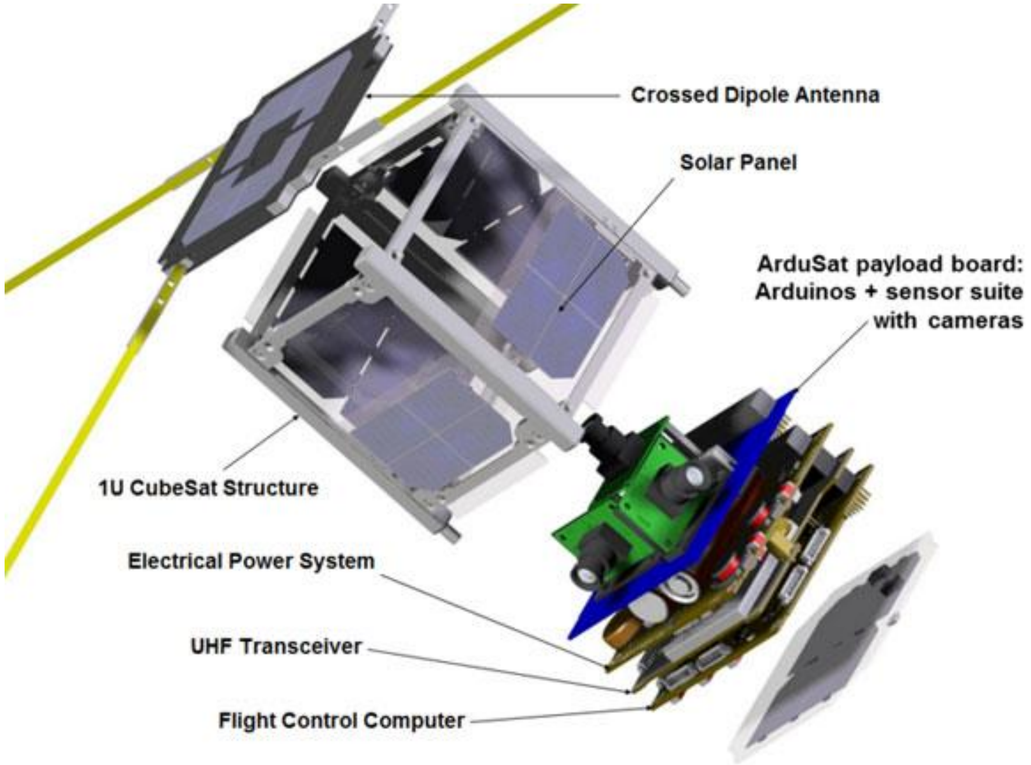
Alguns projetos

Quadricóptero



http://quadricoptero.wordpress.com/

ArduSat

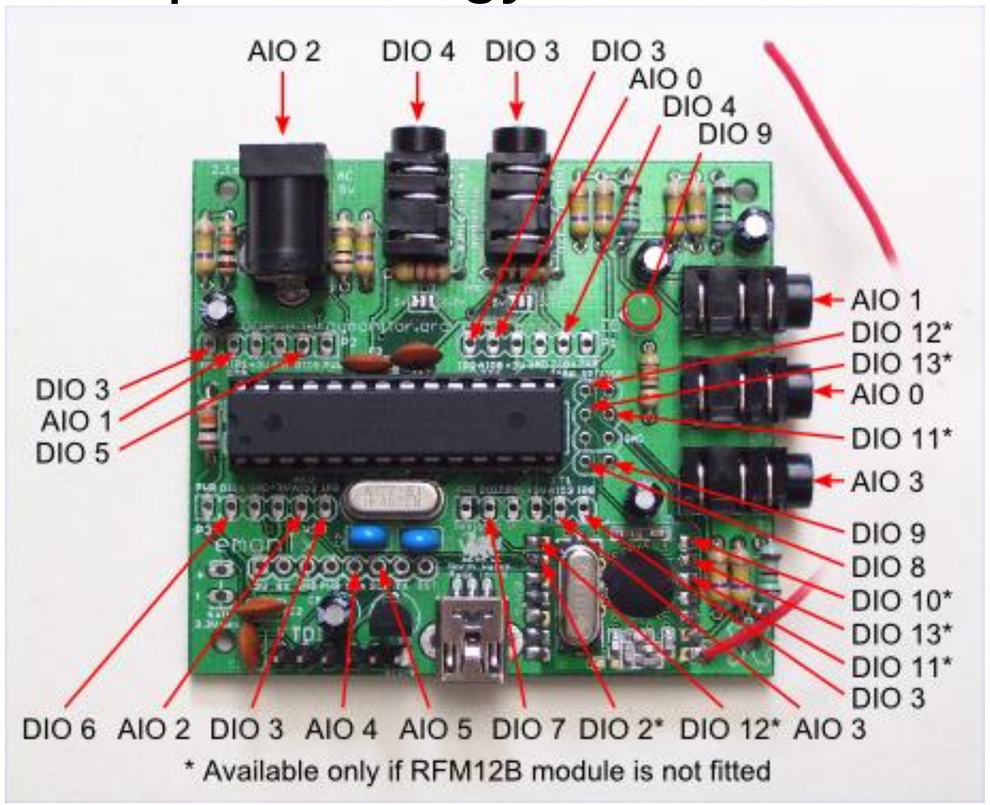


Alguns projetos

Wifi Robot

http://www.jbprojects.net/projects/wifirobot/

Open Energy Monitor



http://openenergymonitor.org

Controlador Lógico Programável

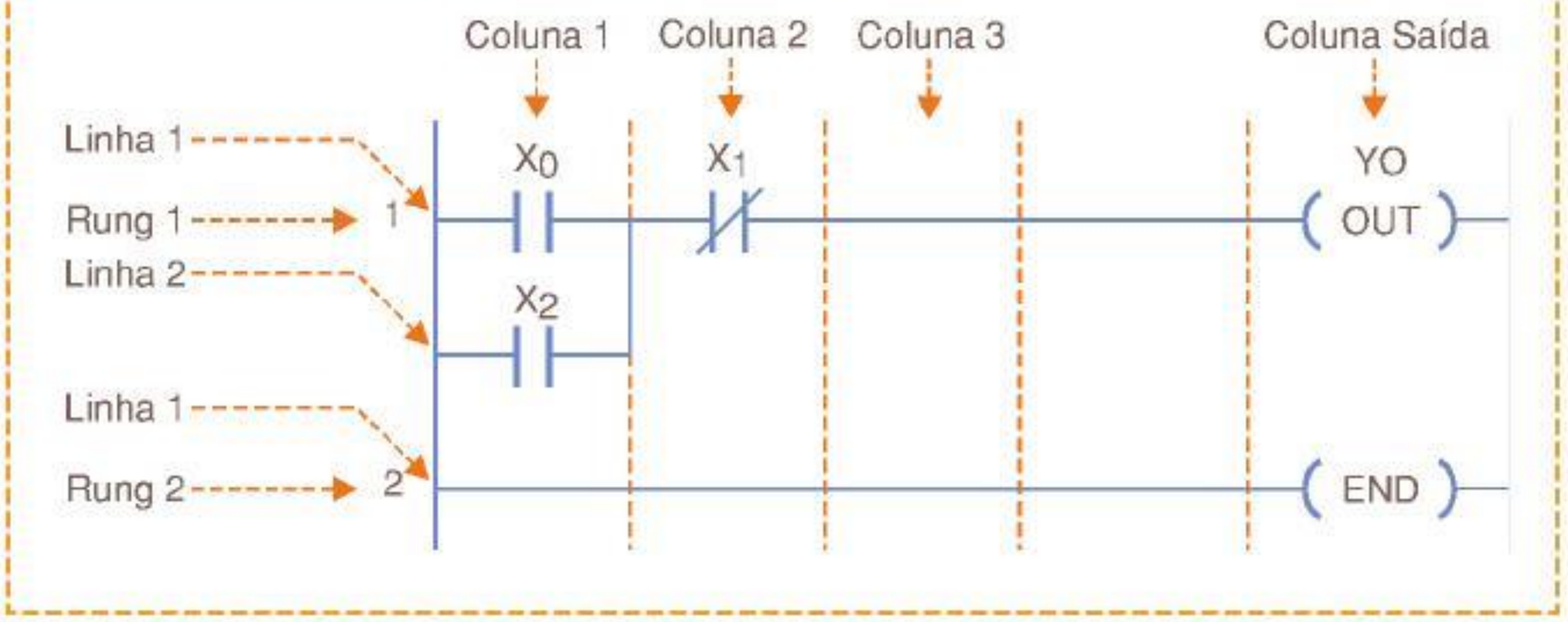
São microcomputadores de propósito específico, dedicado para o controle de processos e equipamentos.

Diferença entre Arduino e CLP

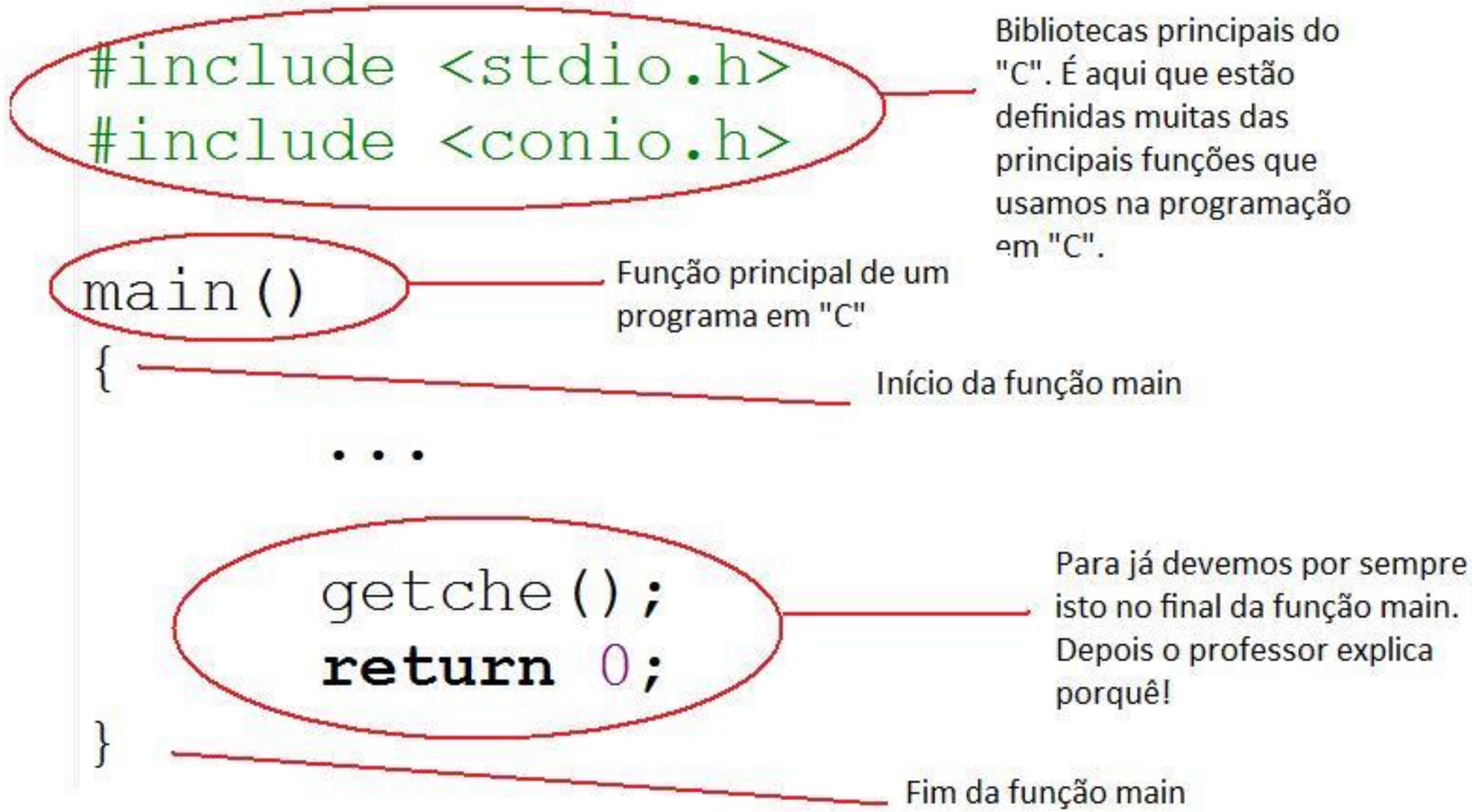
ARDUINO	CLP
Equipamento de proposito geral	Equipamento de proposito especifico
Não contem certificações de segurança, tanto software, quanto hardware. Por ser uma plataforma de desenvolvimento.	Vem com certificações industriais e tem muitas características de segurança.
Faça você mesmo	Varias empresas desenvolvedoras
R$ 60,00 ~ R$ 100,00	A partir de R$300,00

Linguagem de programação - LADDER

1 Representação da linguagem Ladder (escada)



Linguagem de programação – Estruturada “C”



Processo de desenvolvimento - Firmware

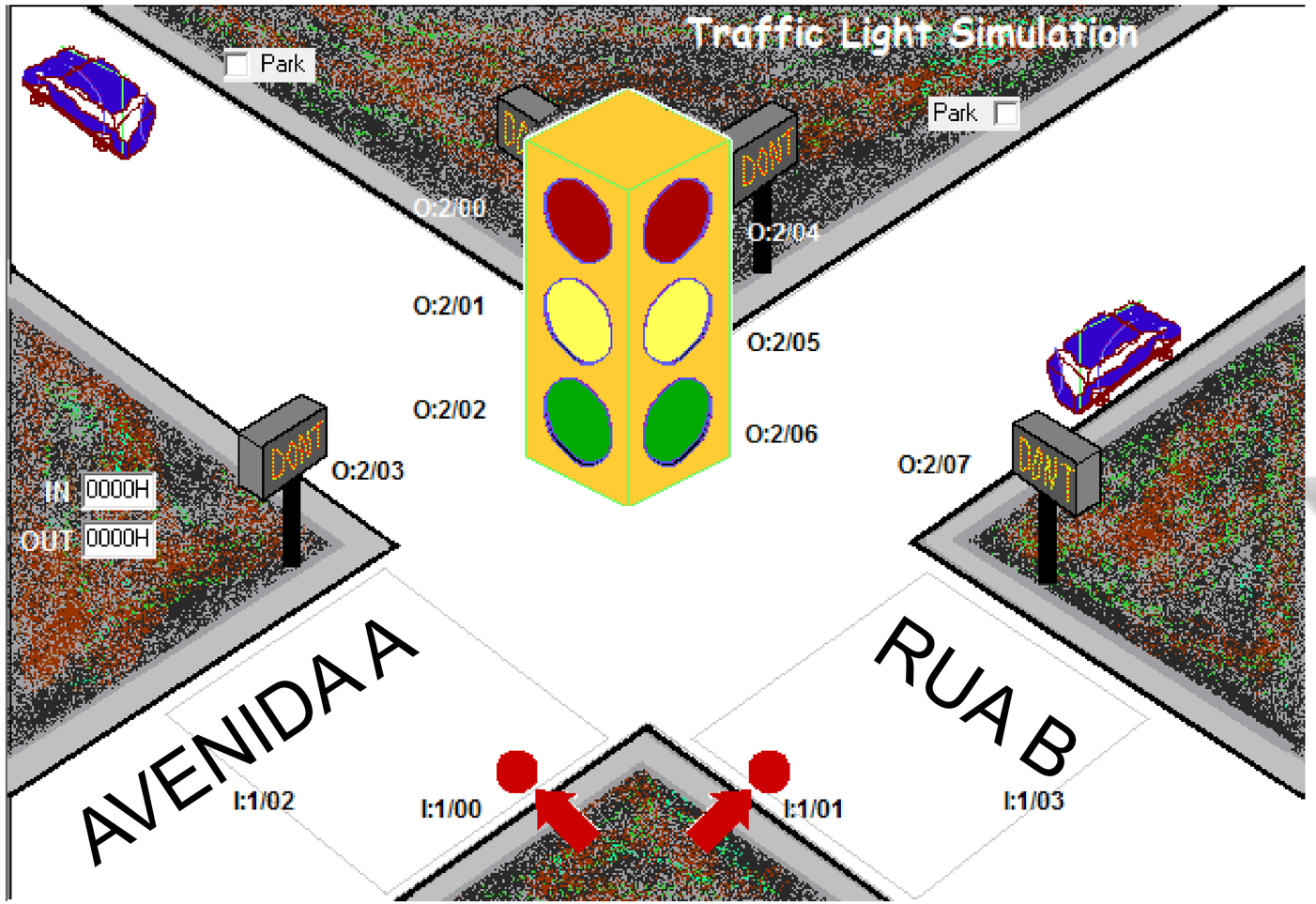
1° • Mapeamento das entradas e saídas do processo

2° • Montagem da maquina de estados

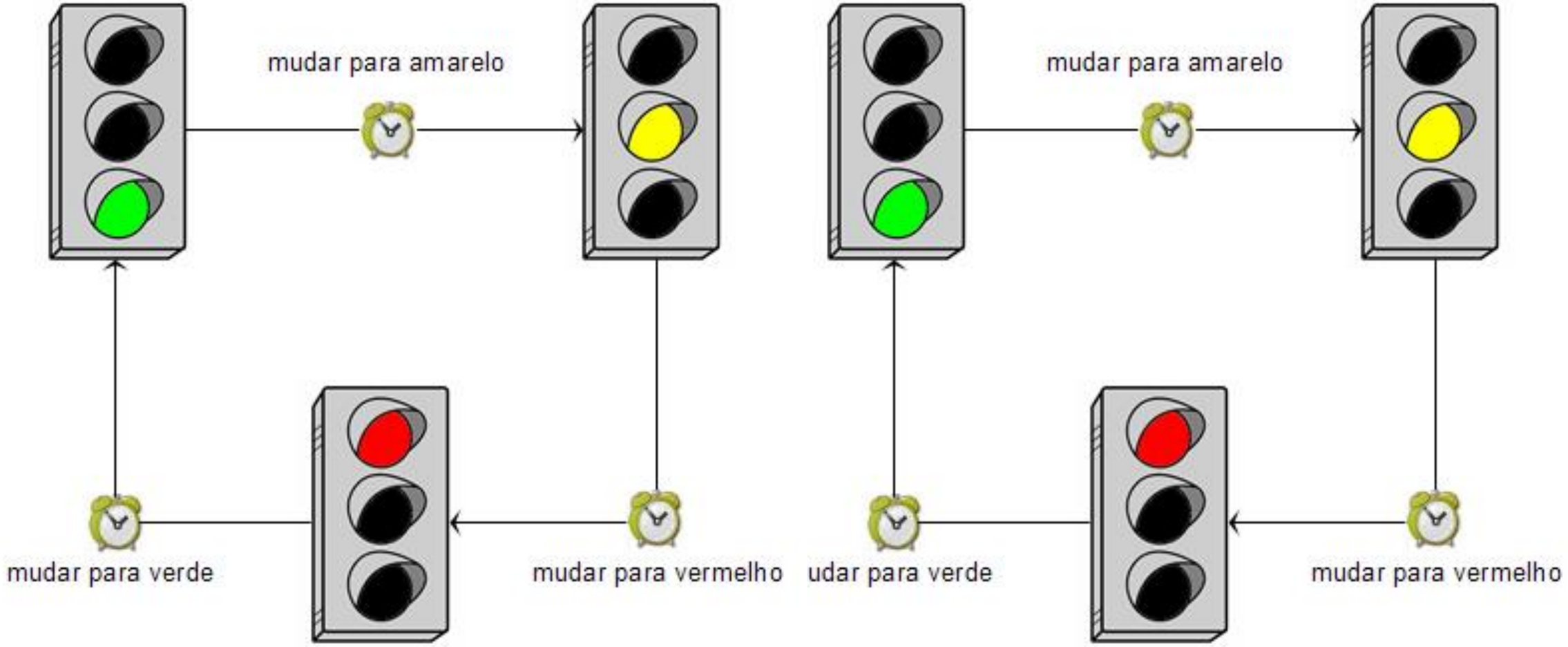
3° • Tabela de transições

4° • Conversão para linguagem de interesse (C ou Ladder)

Processo – Sistema Semaforico



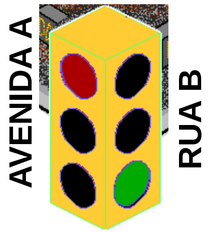
1° Mapeamento das entradas e saídas do processo

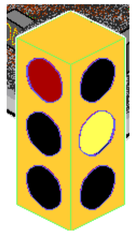


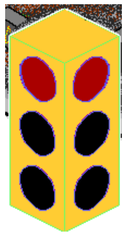
1° Mapeamento das entradas e saídas do processo

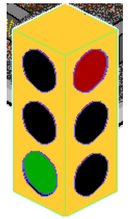
ENTRADAS		SAÍDAS	
NOME	**TIPO**	**NOME**	**TIPO**
TEMP_1	Temporizador	LAMPR1	Digital
TEMP_2	Temporizador	LAMPY1	Digital
TEMP_3	Temporizador	LAMPG1	Digital
TEMP_4	Temporizador	LAMPR2	Digital
TEMP_5	Temporizador	LAMPY2	Digital
TEMP_6	Temporizador	LAMPG2	Digital

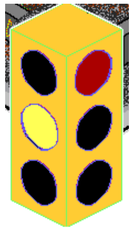
2° Montagem da maquina de estados

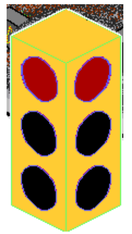












ESTADO 0
- **LAMPR1 = 1**
- LAMPY1 = 0
- LAMPG1 = 0
- LAMPR2 = 0
- LAMPY2 = 0
- **LAMPG2 = 1**

TEMP_1 = 8s

ESTADO 1
- **LAMPR1 = 1**
- LAMPY1 = 0
- LAMPG1 = 0
- LAMPR2 = 0
- **LAMPY2 = 1**
- LAMPG2 = 0

TEMP_2 = 4s

ESTADO 2
- **LAMPR1 = 1**
- LAMPY1 = 0
- LAMPG1 = 0
- **LAMPR2 = 1**
- LAMPY2 = 0
- LAMPG2 = 0

TEMP_3 = 2s

ESTADO 3
- LAMPR1 = 0
- LAMPY1 = 0
- **LAMPG1 = 1**
- **LAMPR2 = 1**
- LAMPY2 = 0
- LAMPG2 = 0

TEMP_4 = 8s

ESTADO 4
- LAMPR1 = 0
- **LAMPY1 = 1**
- LAMPG1 = 0
- **LAMPR2 = 1**
- LAMPY2 = 0
- LAMPG2 = 0

TEMP_5 = 8s

ESTADO 5
- **LAMPR1 = 1**
- LAMPY1 = 0
- LAMPG1 = 0
- **LAMPR2 = 1**
- LAMPY2 = 0
- LAMPG2 = 0

TEMP_6 = 2s

3° Tabela de transições

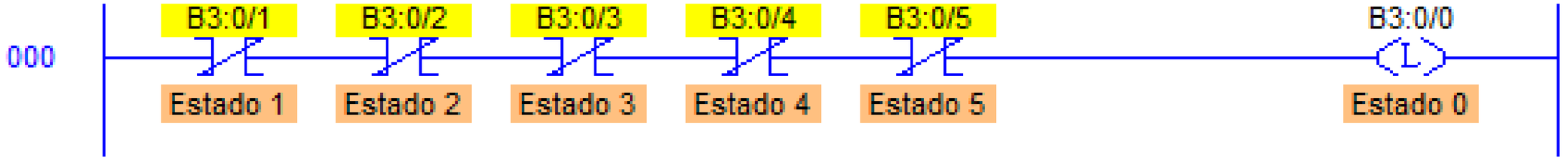
	LAMPR1	LAMPY1	LAMPG1	LAMPR2	LAMPY2	LAMPG2
ESTADO 0	1	0	0	0	0	1
ESTADO 1	1	0	0	0	1	0
ESTADO 2	1	0	0	1	0	0
ESTADO 3	0	0	1	1	0	0
ESTADO 4	0	1	0	1	0	0
ESTADO 5	1	0	0	1	0	0

4. Conversão para a linguagem de interesse

- 1º Inicialização do Estado 0
- 2º Executa o processo de transição
- 3º Execução das ações

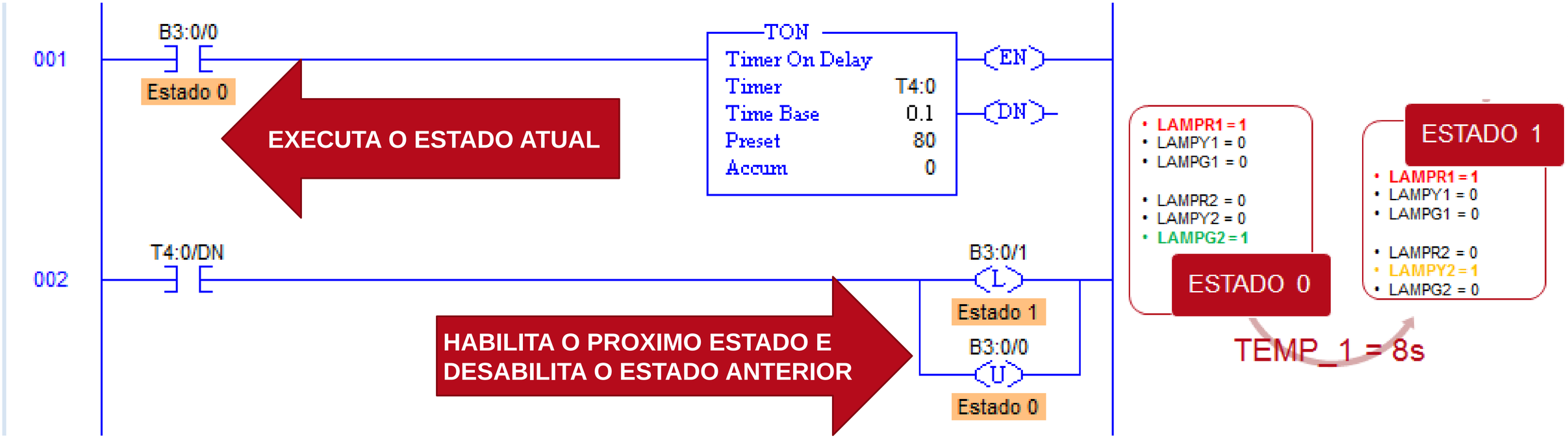
4.1 Conversão para a linguagem Ladder

1º Inicialização do Estado 0



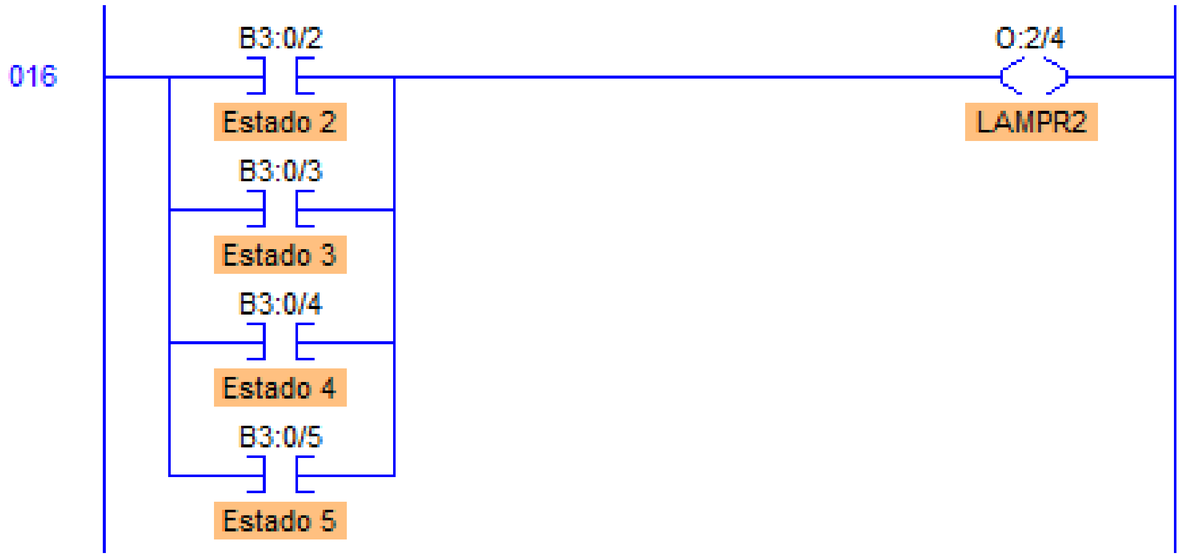
4.1 Conversão para a linguagem Ladder

2º Executa o processo de transição



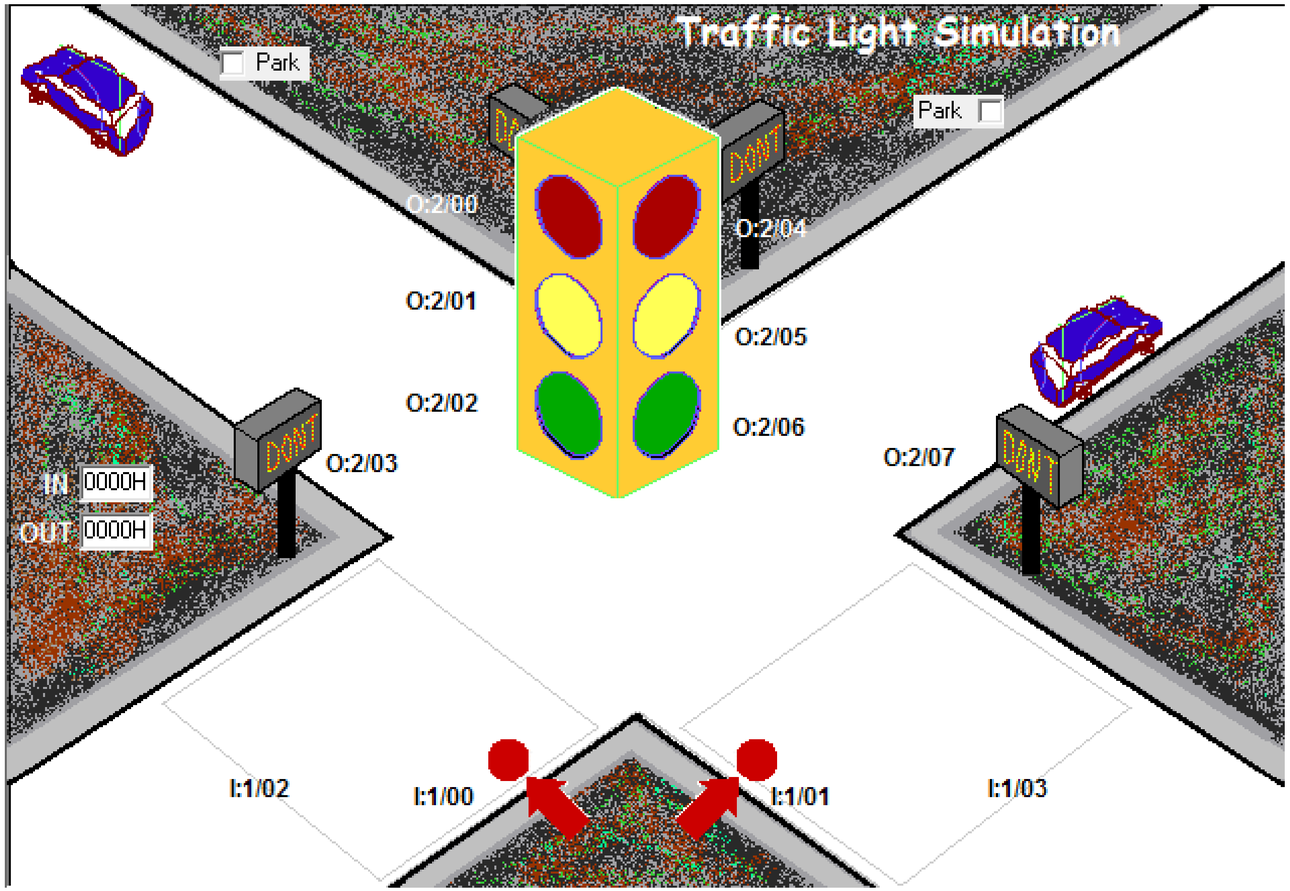
4.1 Conversão para a linguagem Ladder

3º Execução das ações



	LAMPR1	LAMPY1	LAMPG1	LAMPR2	LAMPY2	LAMPG2
ESTADO 0	1	0	0	0	0	1
ESTADO 1	1	0	0	0	1	0
ESTADO 2	1	0	0	1	0	0
ESTADO 3	0	0	1	1	0	0
ESTADO 4	0	1	0	1	0	0
ESTADO 5	1	0	0	1	0	0

Testando...


Traffic Light Simulation
Park
Park
DONT
DONT
DONT
O:2/00
O:2/01
O:2/02
O:2/03
O:2/04
O:2/05
O:2/06
O:2/07
IN 0000H
OUT 0000H
I:1/02
I:1/00
I:1/01
I:1/03

4.2 Conversão para a linguagem C

1º Inicialização do Estado 0

```
/* FUNÇÃO DE INICIALIZAÇÃO */
void initSM(){

    estadoatual = EST_0;
    acionamentoLampadas(0,0,0,0,0,0);

}
```

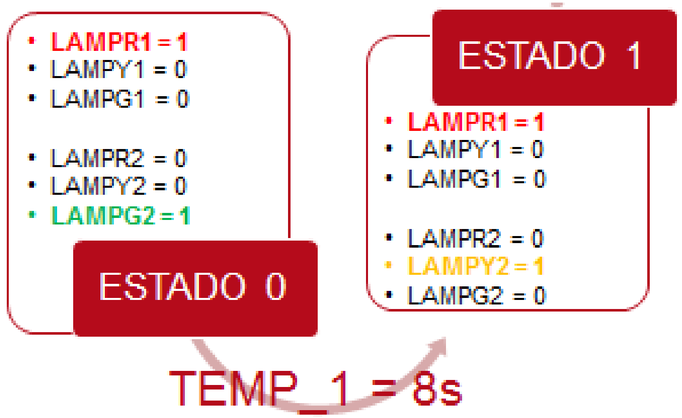
4.2 Conversão para a linguagem C

2º Executa o processo de transição

```
switch (estadoatual) {

    case EST_0:

        acionamentoLampadas(1,0,0,0,0,1);
        delay(TEMP8);
        estadoatual = EST_1;

    break;

    case EST_1:

        acionamentoLampadas(1,0,0,0,1,0);
        delay(TEMP4);
        estadoatual = EST_2;

    break;
```

EXECUTA O ESTADO ATUAL

HABILITA O PROXIMO ESTADO E DESABILITA O ESTADO ANTERIOR



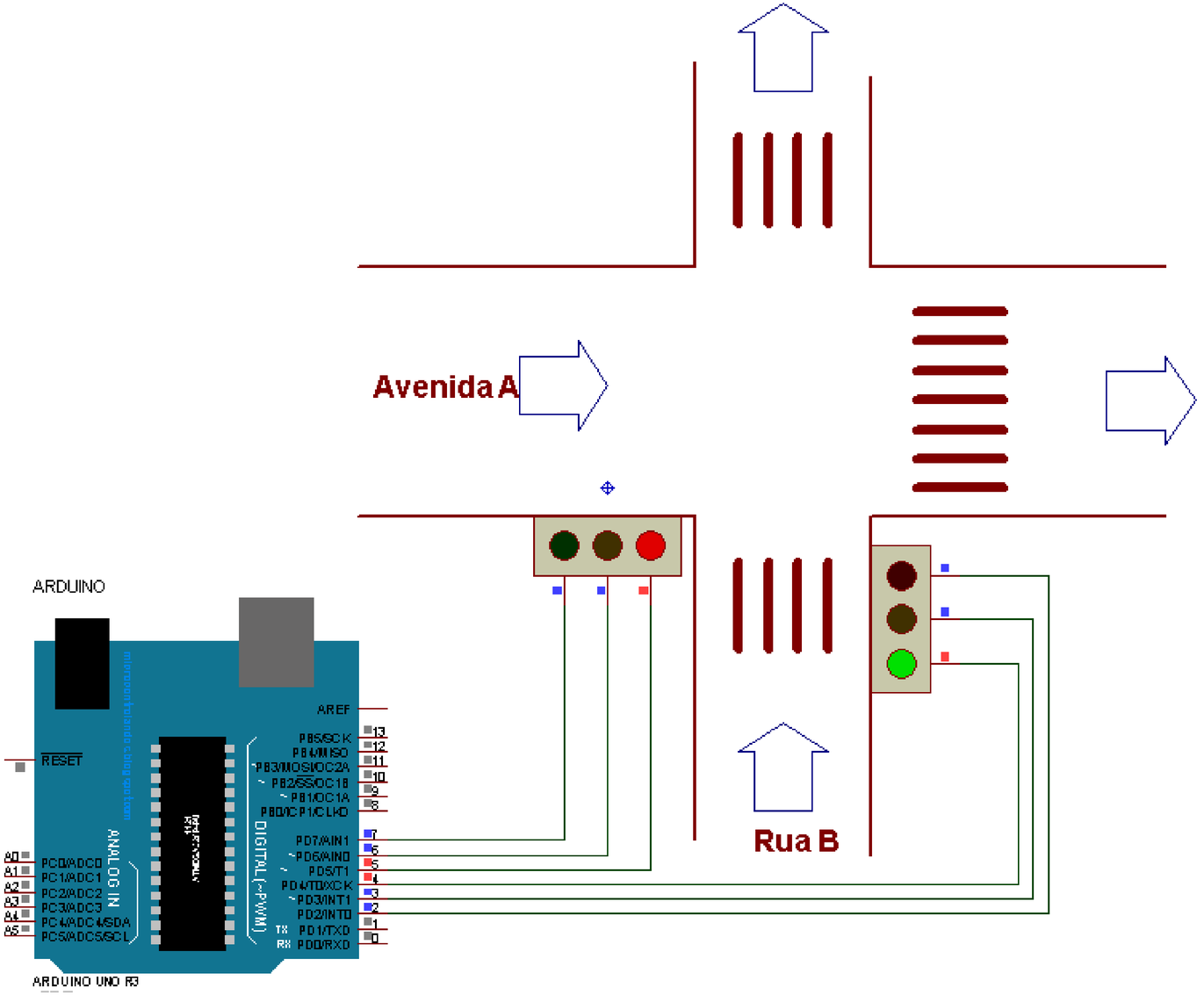
4.2 Conversão para a linguagem C

3º Execução das ações

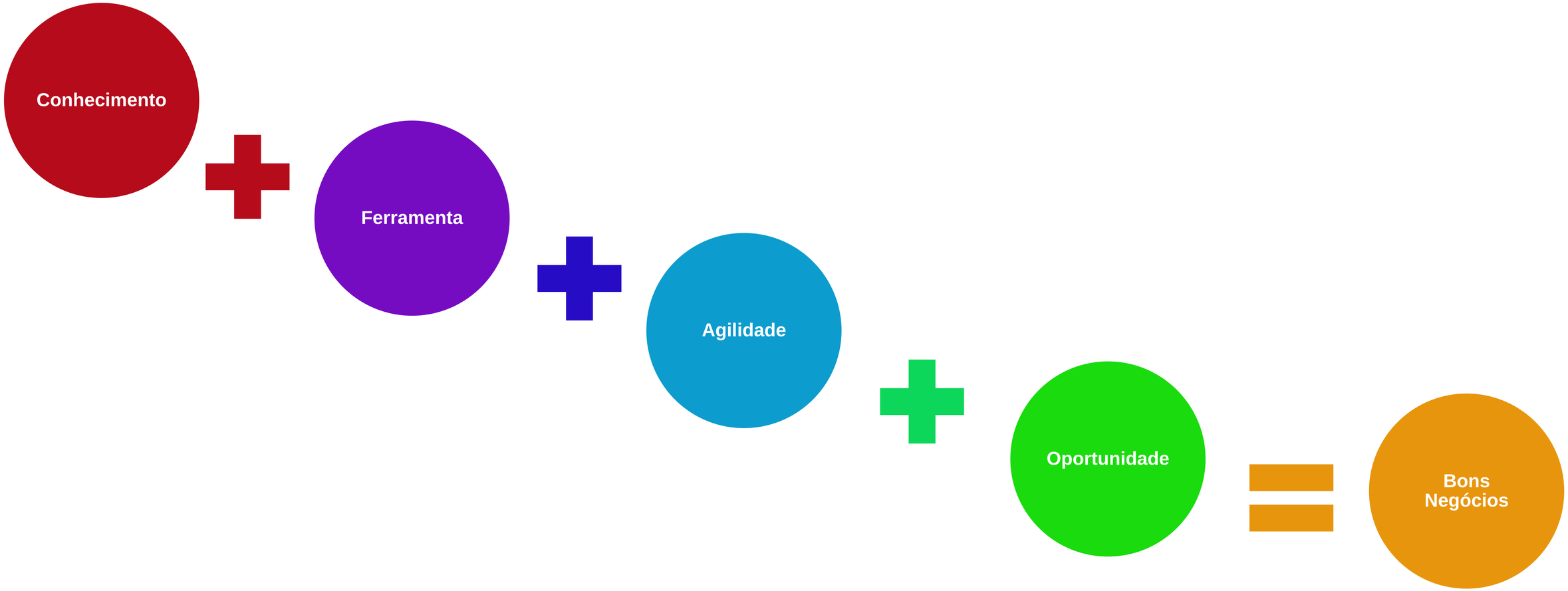
```
/* FUNÇÕES DE CONTROLE */
void acionamentoLampadas(byte vbLampR1,byte vbLampG1,byte
                         vbLampY1,byte vbLampR2,byte vbLampG2,
                         byte vbLampY2){

        digitalWrite(LAMPR1,vbLampR1);
        digitalWrite(LAMPY1,vbLampY1);
        digitalWrite(LAMPG1,vbLampG1);
        digitalWrite(LAMPR2,vbLampR2);
        digitalWrite(LAMPY2,vbLampY2);
        digitalWrite(LAMPG2,vbLampG2);
}
```

	LAMPR1	LAMPY1	LAMPG1	LAMPR2	LAMPY2	LAMPG2
ESTADO 0	1	0	0	0	0	1
ESTADO 1	1	0	0	0	1	0
ESTADO 2	1	0	0	1	0	0
ESTADO 3	0	0	1	1	0	0
ESTADO 4	0	1	0	1	0	0
ESTADO 5	1	0	0	1	0	0

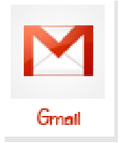
Testando...


Avenida A
Rua B
ARDUINO
ARDUINO UNO R3
RESET
AREF
ANALOG IN
DIGITAL (~PWM)
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
PB5/SCK
PB4/MISO
~PB3/MOSI/OC2A
~PB2/SS/OC1B
~PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~PD6/AIN0
~PD5/T1
PD4/T0/XCK
~PD3/INT1
PD2/INT0
TX PD1/TXD
RX PD0/RXD

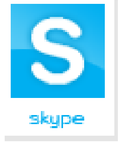
Conclusão


Conhecimento
Ferramenta
Agilidade
Oportunidade
Bons
Negócios

Obrigado!



hjssena@gmail.com



hamilton.sena

http://hamiltonsena.net