

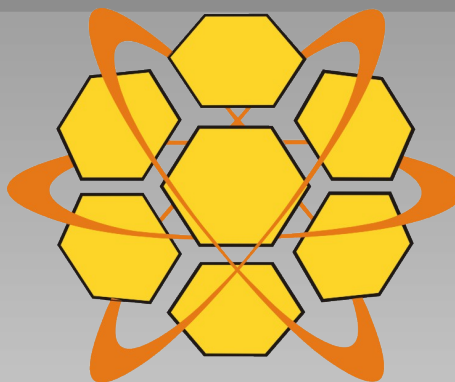


# MINICURSO ARDUINO

Ministrantes:

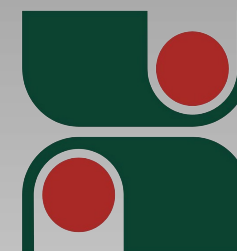
DANIEL S. CAMARGO

GIAN LUCAS NUNES



**COLMÉIA**

Grupo de Pesquisa em  
Software e Hardware Livre



**UDESC**

Joinville

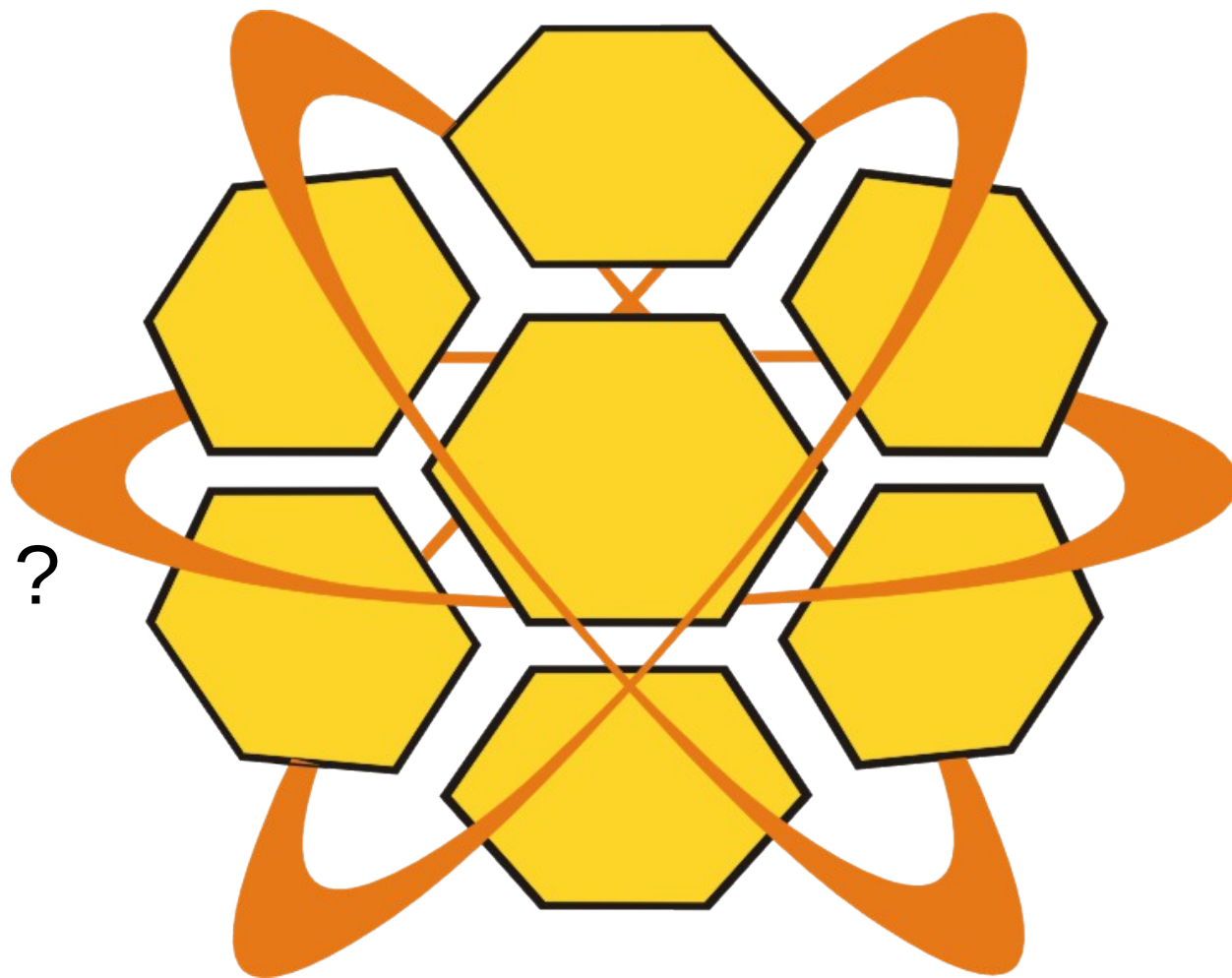
Material Disponível no site:

[www.colmeia.udesc.br](http://www.colmeia.udesc.br)



# COLMÉIA

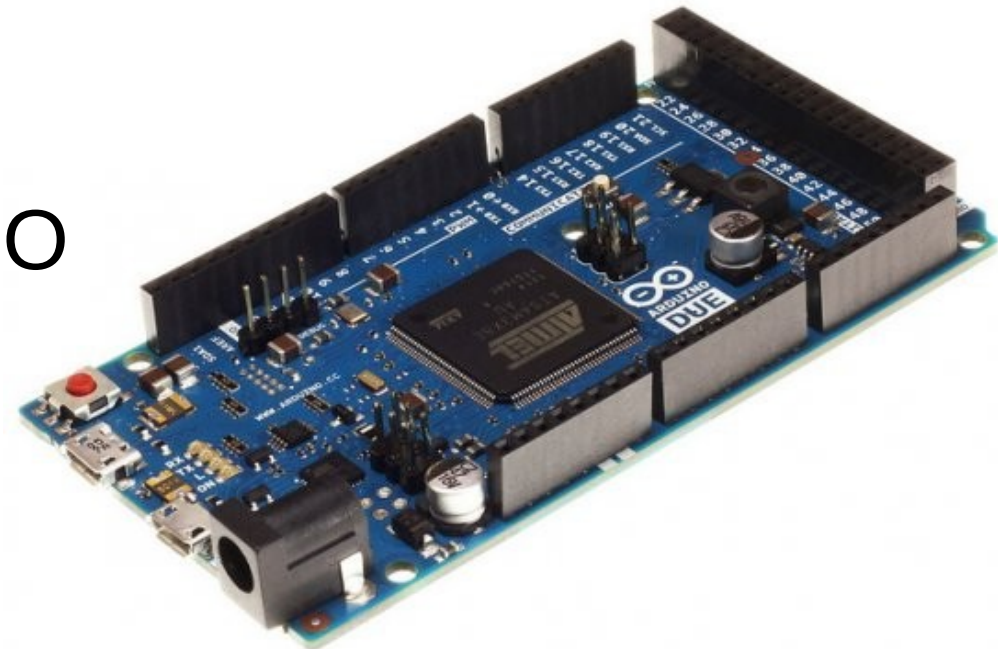
- Quem somos?
- O que fazemos ?



Contato : [www.colmeia.udesc.br](http://www.colmeia.udesc.br)



# PLATAFORMA DE PROTOTIPAGEM ARDUINO

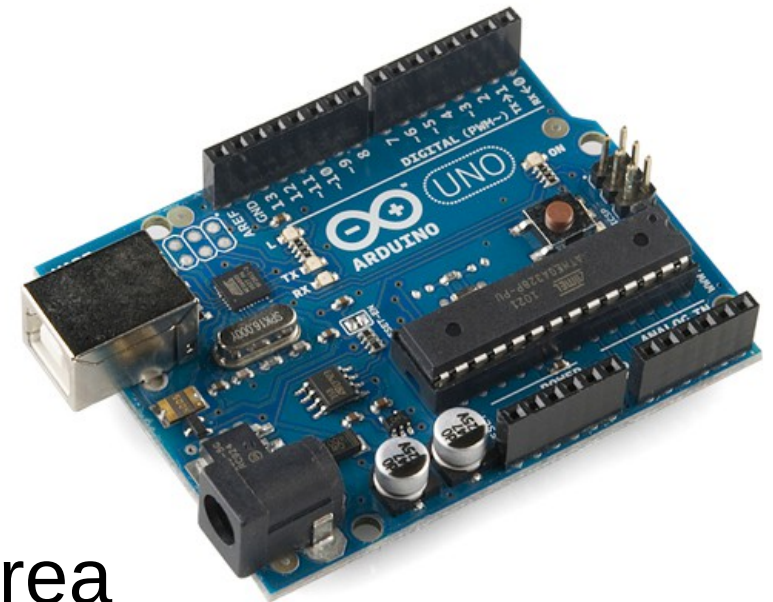


Arduino Due  
Fonte: <http://store.arduino.cc>



# ARDUINO

- Criado por Massimo Banzi
- Projeto iniciado em 2005
- Interaction Design Institute Ivrea
- Palavra de origem germânica que significa Grande Amigo



Arduino UNO

Fonte: <http://store.arduino.cc>



# ARDUINO

- É um conjunto de ferramentas que possibilitam o desenvolvimento de dispositivos eletrônicos;
- Plataforma de prototipagem com software e hardware flexíveis e fáceis de usar;



Arduino Duemilanove

Fonte:  
<http://australianrobotics.com.au/>





# O HARDWARE

NO DUEMILANOVE:

- Microcontrolador: ATmega328 (32k flash)
- 14 pinos digitais (entradas ou saídas)
- 6 pinos de PWM
- 6 pinos analógicos
- 1 pino e um botão de RESET
- 2 Pinos de alimentação
- Uma entrada USB e uma Entrada de alimentação 5V



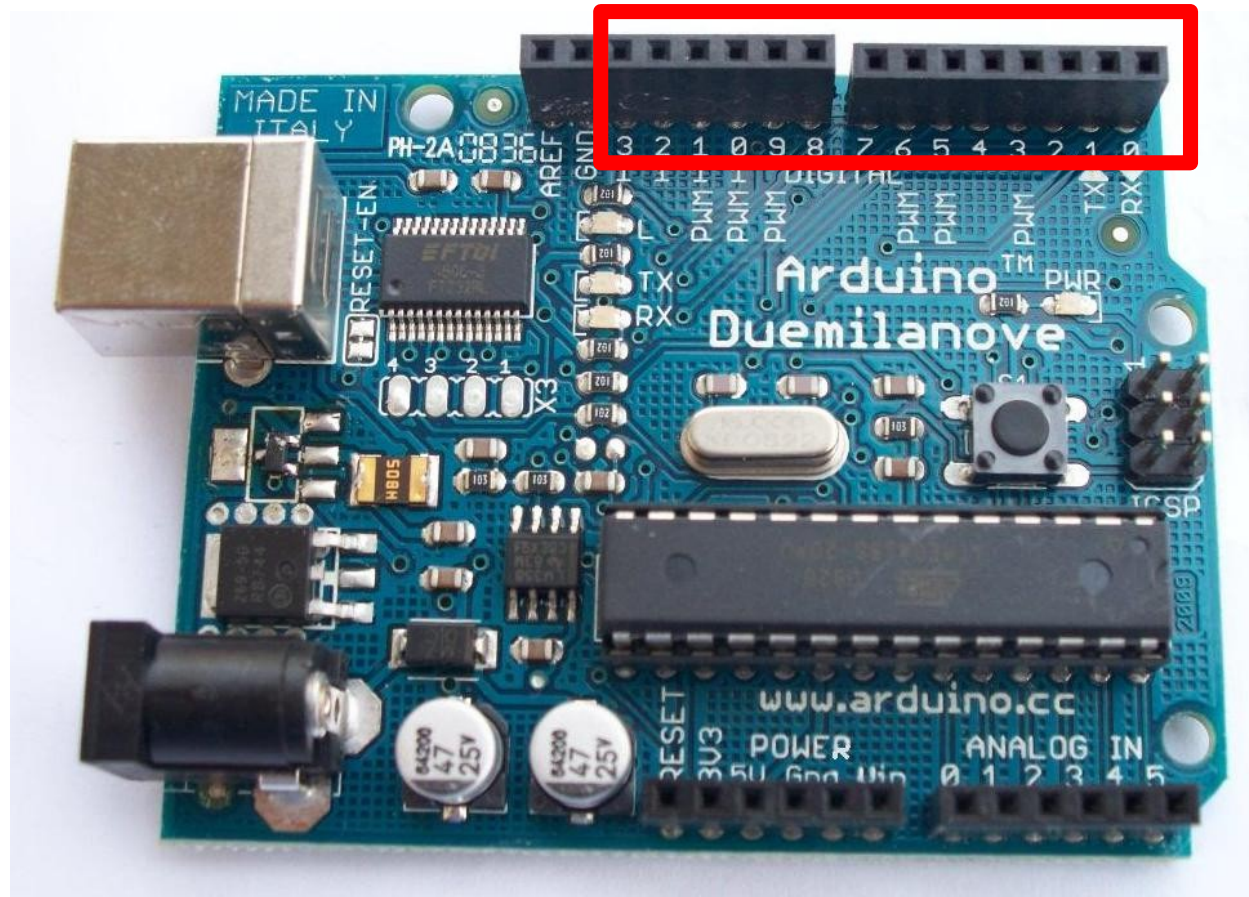
ATmega328

Fonte: [multilogica-shop.com](http://multilogica-shop.com)



# DISPOSIÇÃO DO HARDWARE

Entradas e saídas Digitais (I/O)

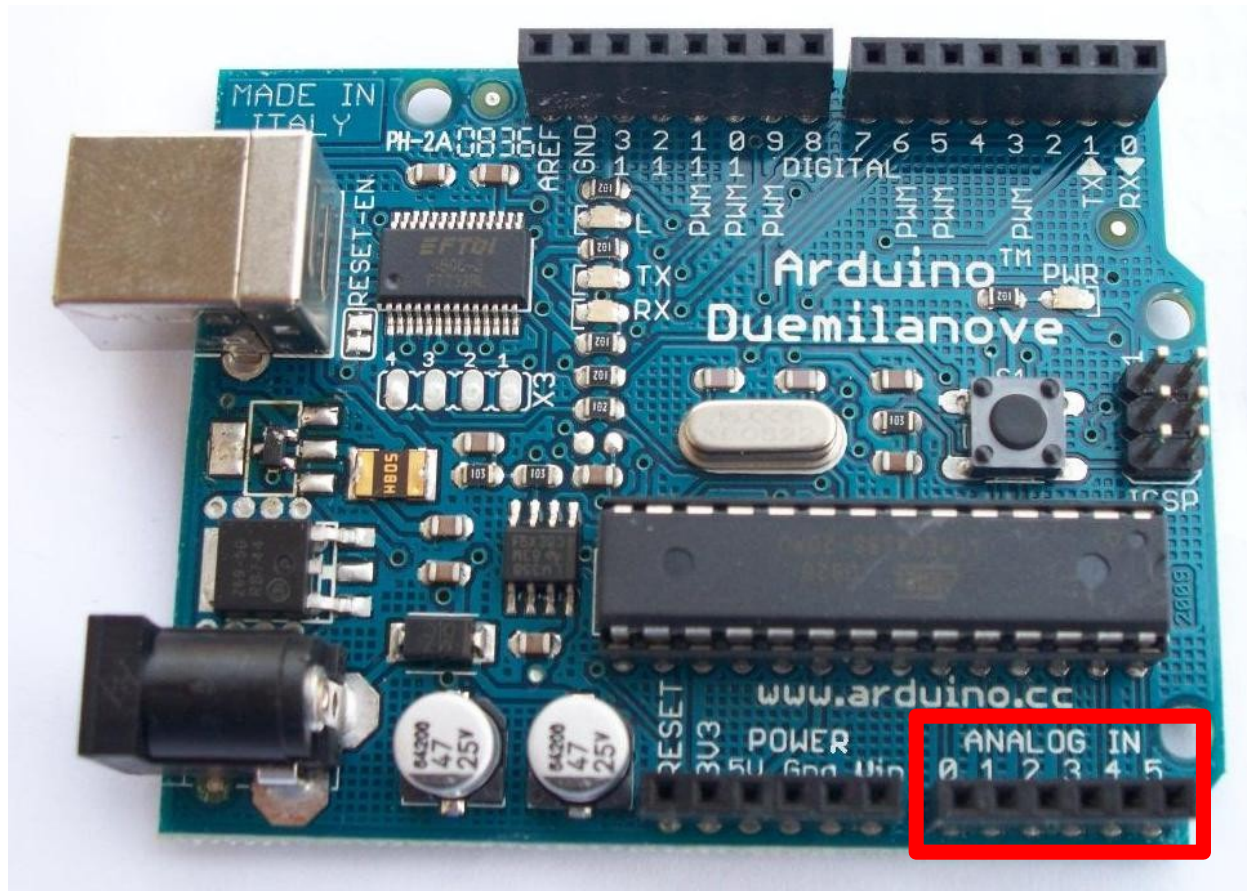


Fonte: <http://www.reuk.co.uk/wordpress/standard-alone-arduino-on-a-breadboard/>



# DISPOSIÇÃO DO HARDWARE

## Entradas Analógicas



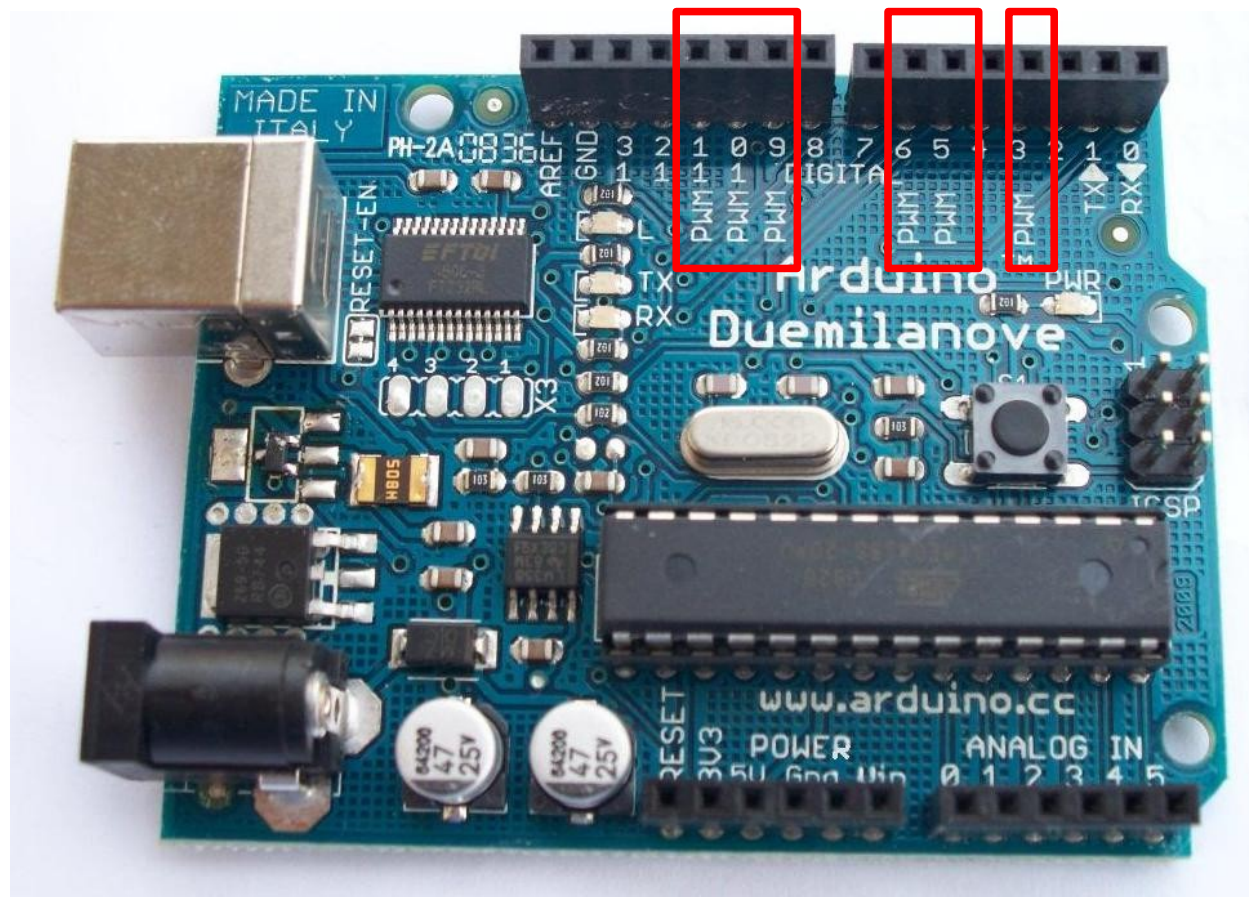
Fonte: <http://www.reuk.co.uk/wordpress/standard-alone-arduino-on-a-breadboard/>





# DISPOSIÇÃO DO HARDWARE

## Pinos de PWM

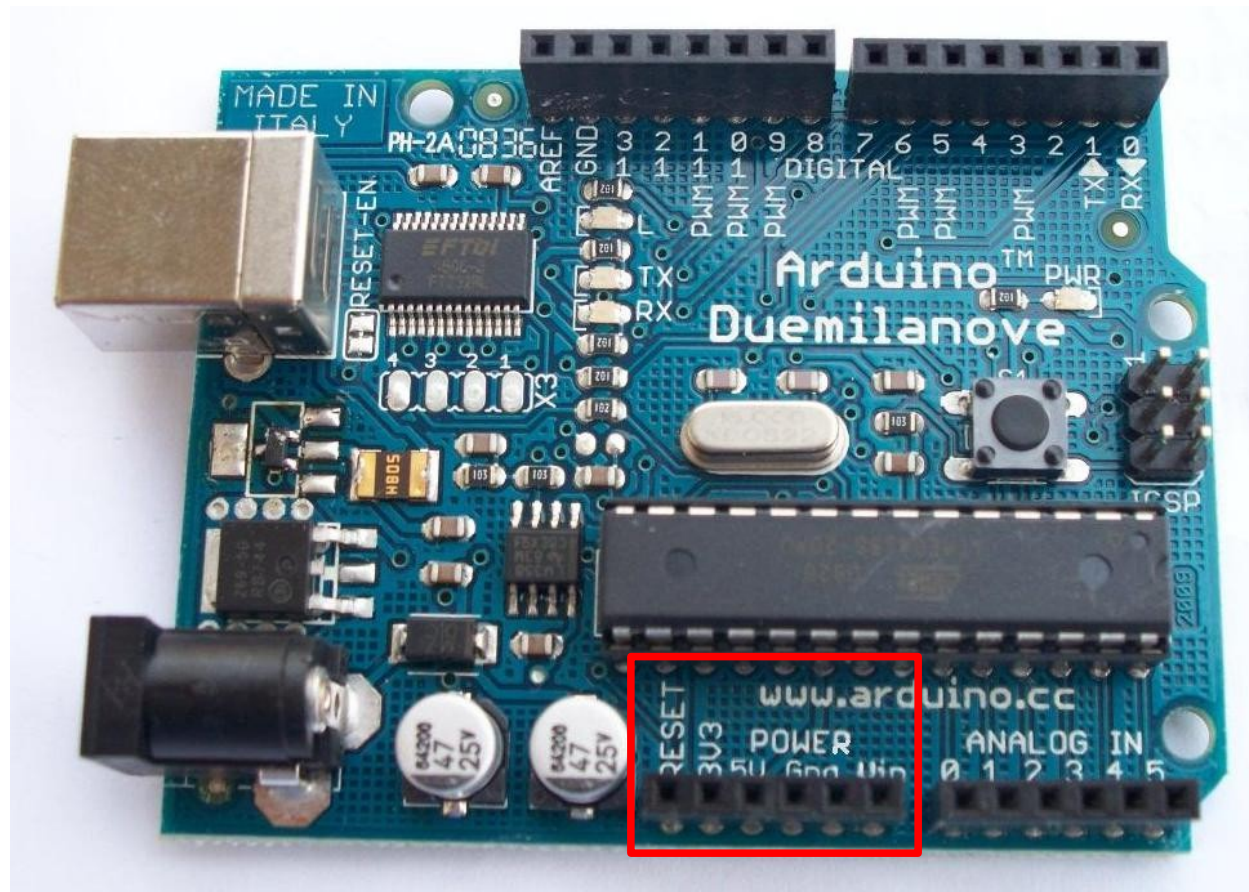


Fonte: <http://www.reuk.co.uk/wordpress/standard-alone-arduino-on-a-breadboard/>

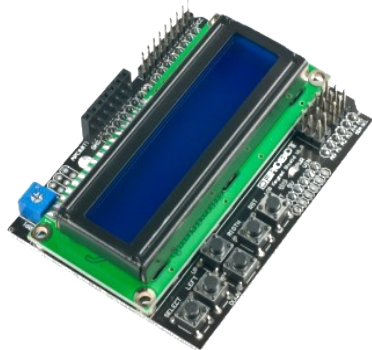


# DISPOSIÇÃO DO HARDWARE

## Pinos de alimentação



Fonte: <http://www.reuk.co.uk/wordpress/standard-alone-arduino-on-a-breadboard/>



# Vantagens da Plataforma: Shields

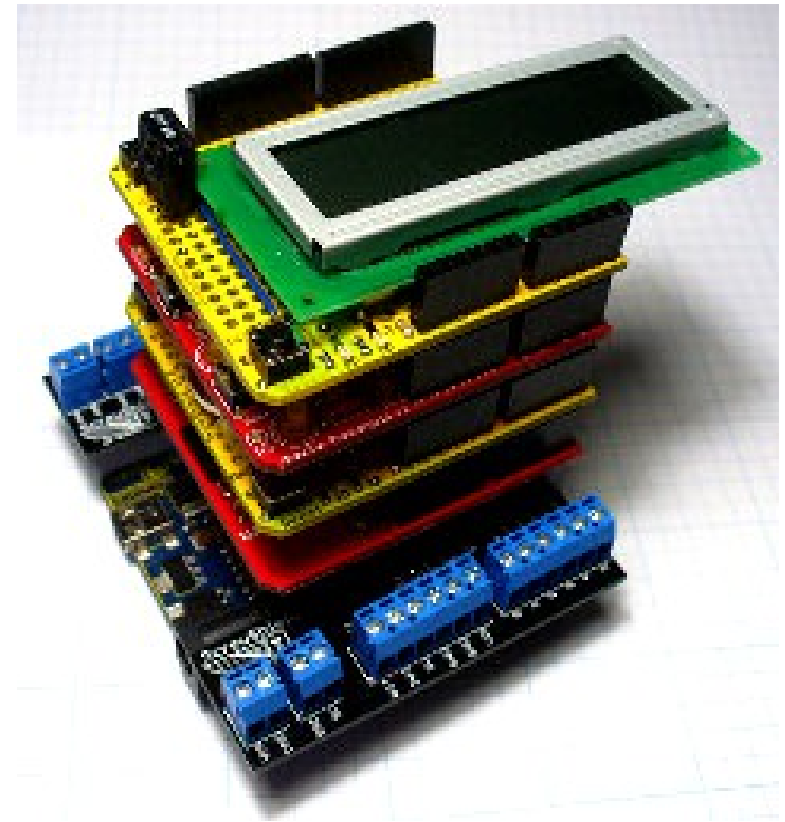
Fonte:  
<http://robosavvy.com/>





# Shields: O que é?

- Placas de circuito impresso com dispositivos de entrada/saída normalmente fixados no topo do aparelho;
- Atualmente, existem mais de 288 variações comerciais em [shieldlist.org](http://shieldlist.org).



Pilha exagerada  
Fonte: [shieldlist.org](http://shieldlist.org)





# Shields: Por que usar?

- Projeto complexo?  
Utilize uma Shield e usufrua de bibliotecas prontas;
- Dá um toque profissional ao projeto; reduz espaço; ótimo para quem não possui conhecimento técnico;



LCD Touch Shield  
Fonte: shieldlist.org



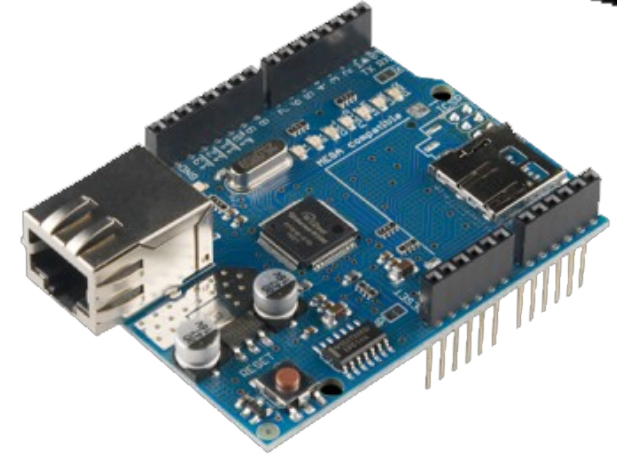
# Shields: Exemplos



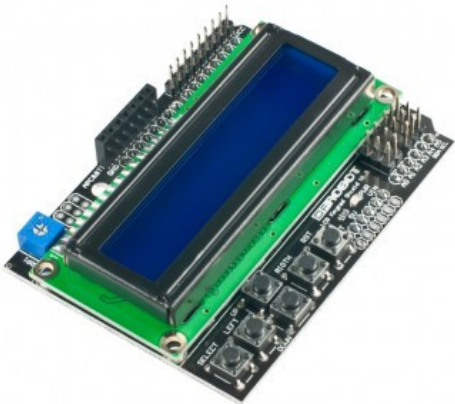
GPS



Xbee



Ethernet+SD



LCD



Motor



RFID

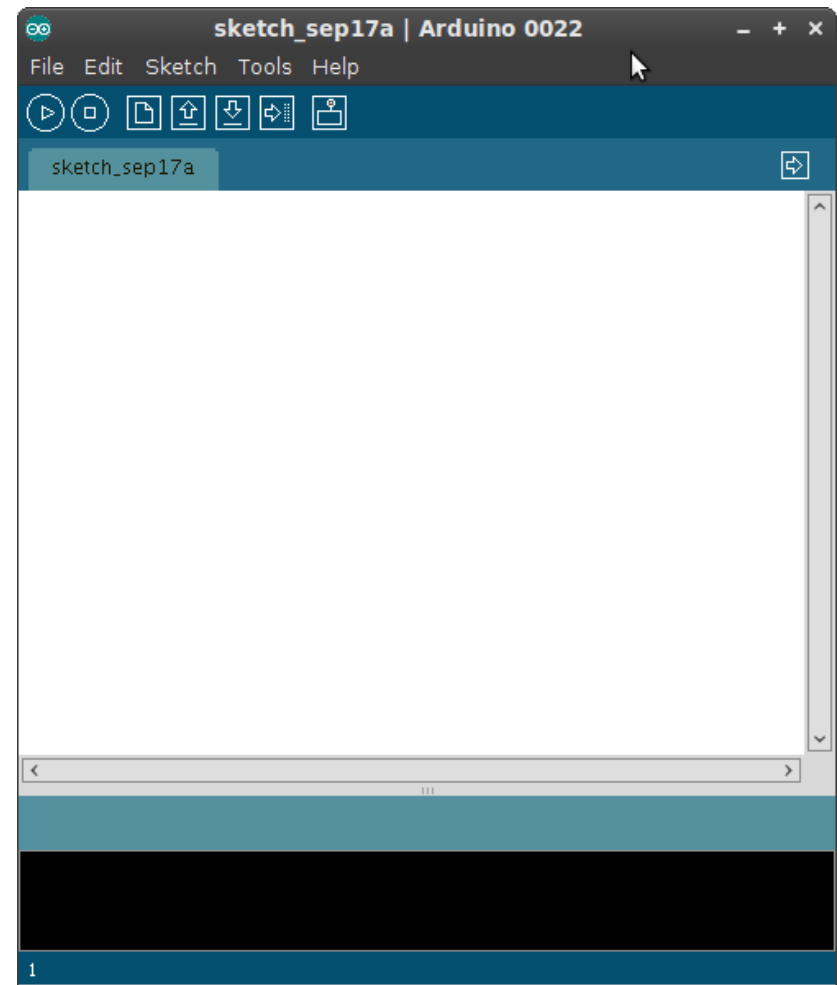


# ARDUINO IDE

(Ambiente Integrado de Desenvolvimento)



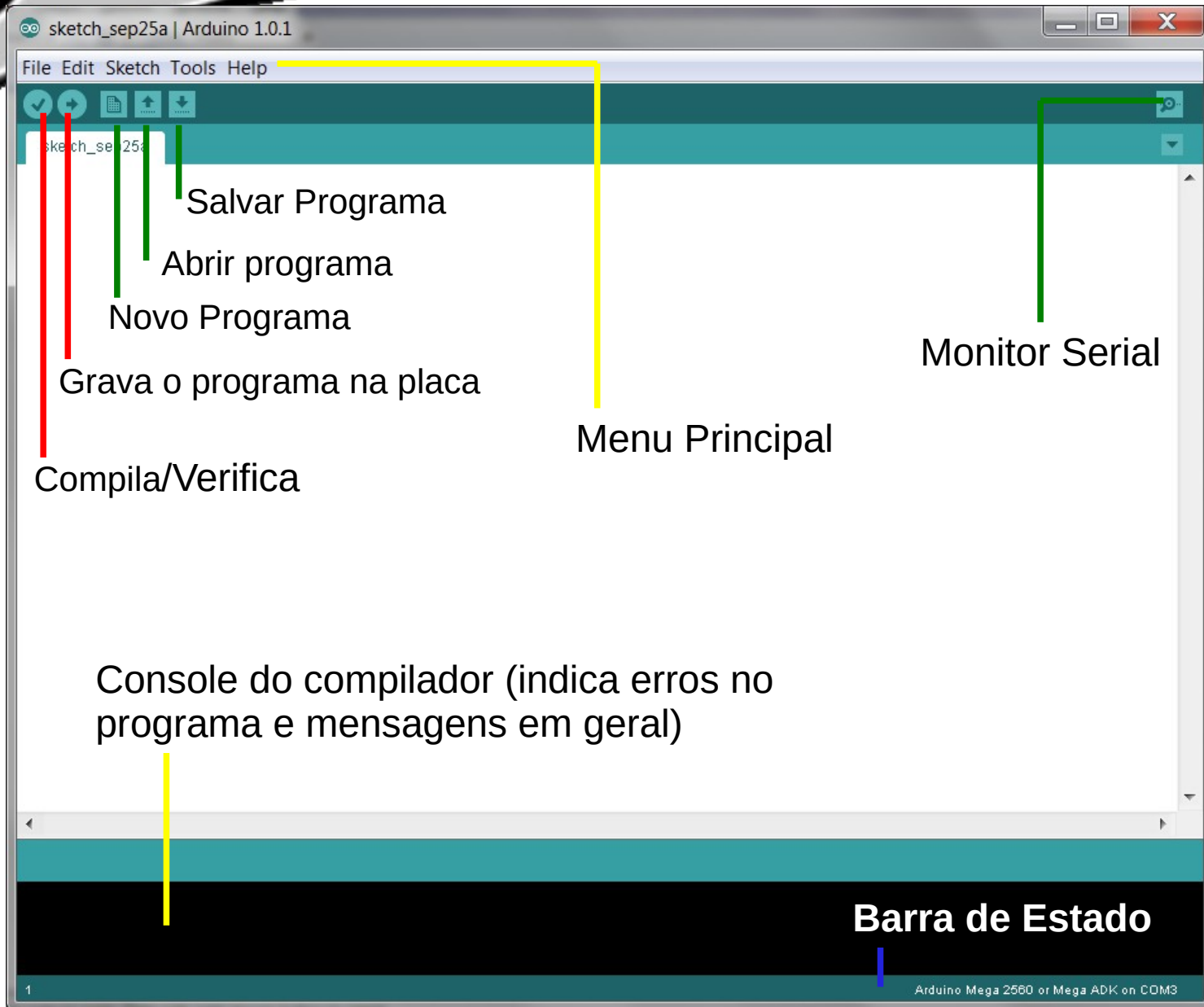
Fonte: [www.varesano.net](http://www.varesano.net)



Print Screen IDE Arduino v022



# Detalhes IDE



Print IDE  
Arduino  
v1.0.1  
Autorial





# Sintaxe do Programa

```
void setup(){  
...  
}
```

Configuração dos pinos como entrada ou saída Executado somente uma vez início do programa

```
void loop(){  
...  
}
```

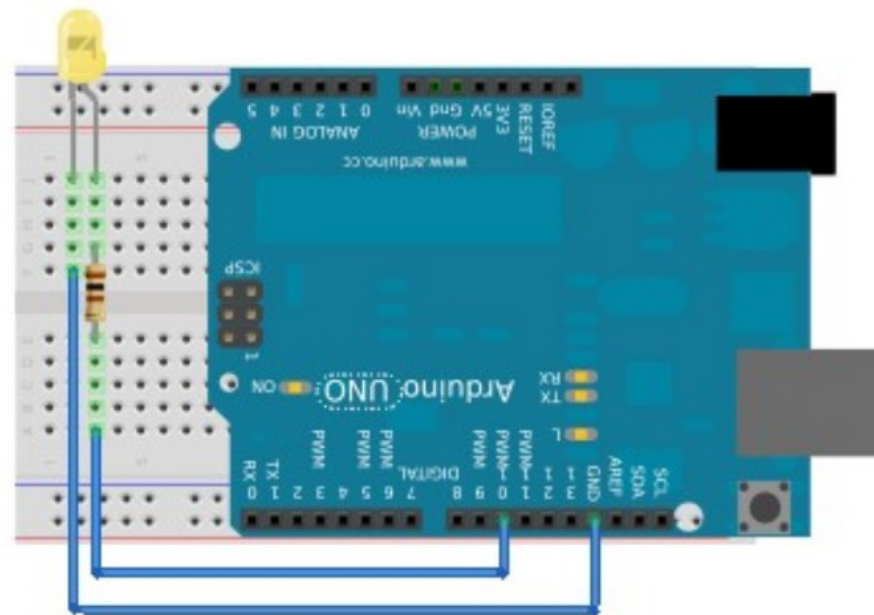
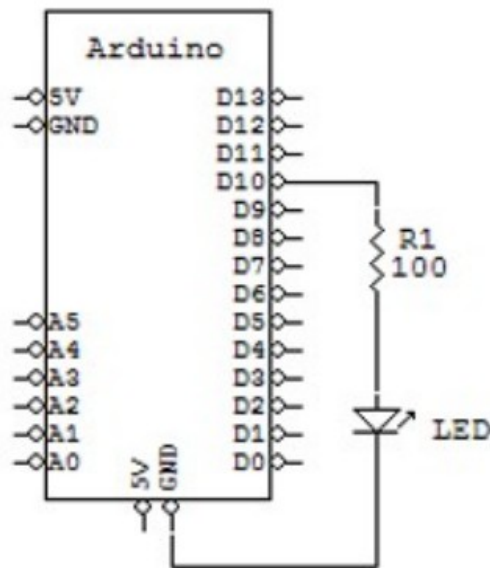
Descreve a rotina a ser executada pelo microcontrolador

Linguagem Baseada em C/C++ logo sua programação é Case Sensitive



# Projeto 1 – Piscar LED

## Esquemático e Montagem Física



Sketch feito com Fritzing



# Projeto 1 – Pisca LED

## FUNÇÕES IMPORTANTES PARA ESTE PROJETO

- **pinMode**(pino, modo); Configura um pino específico para se comportar como uma entrada ou uma saída.
- **digitalWrite**(pino, modo); Escreve em um pino digital o valor lógico Alto ou Baixo.
- **delay**(tempo); Pausa o programa durante um certo tempo especificado em milissegundos.



# Projeto 1 – PISCAR LED

- 1º ETAPA: Declarar Variáveis
- 2º ETAPA: Configurar saídas (void setup( ))
- 3º ETAPA: Lógica do programa (void loop( ))
- 4º ETAPA: Compilar e enviar para o Arduino





# Projeto 1 – Pisca LED

```
int LED = 10;

void setup() { //setup lido na inicialização
  pinMode(LED, OUTPUT); // inicializa o pino como saída
}

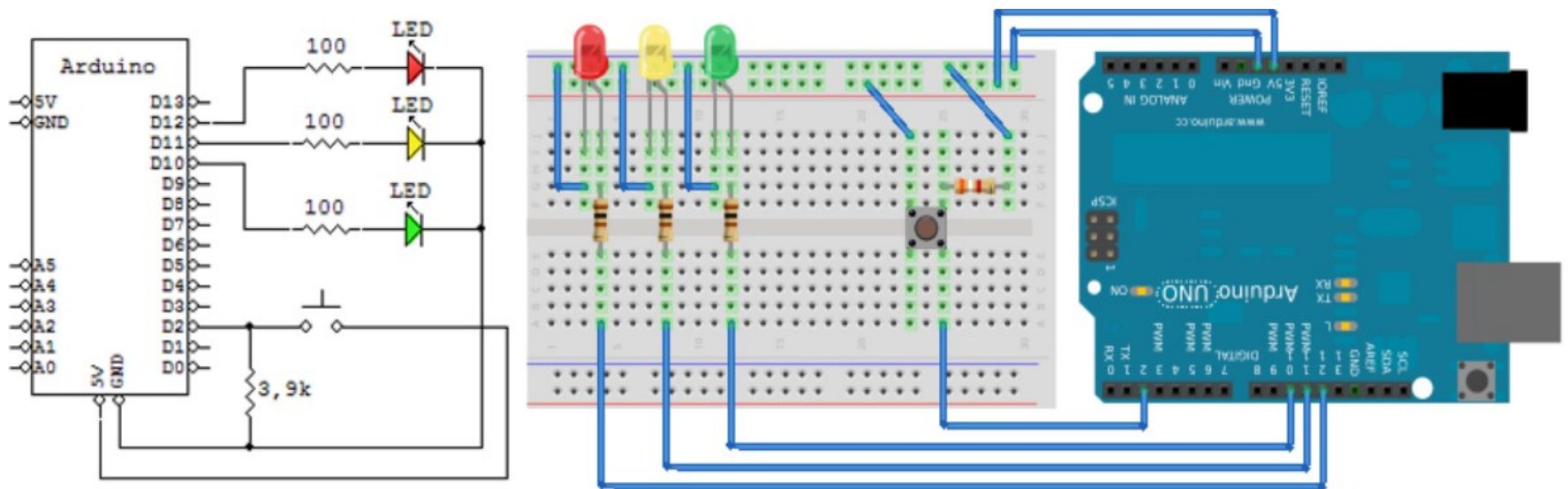
void loop() { // loop executa repetidamente
  digitalWrite(LED, HIGH); // liga o LED
  delay(1000); // espera 1 segundo
  digitalWrite(LED, LOW); // desliga o LED
  delay(1000); // espera 1 segundo
}
```



# Projeto 2 - Semáforo

OBJETIVO: Projetar um semáforo com 3 LEDs e um botão para que os pedestres possam atravessar.

## Esquemático e Hardware



Sketch feito com Fritzing



# Projeto 2 - Semáforo

## FUNÇÕES IMPORTANTES PARA O PROJETO

- **attachInterrupt**(canal, função, modo); Função que habilita alguma interrupção externa.
  - Canal : 0 para o pino 2 e 1 para o pino 3
  - Função: será executada pela interrupção
  - Modo: tipo da interrupção (**LOW**, **CHANGE**, **RISING** e **FALLING**)
- **pinMode**(pino, modo);
- **digitalWrite**(pino, modo);
- **delay**(tempo);



# Projeto 2 - Semáforo

- 1º PASSO: Declarar variáveis
- 2º PASSO: Configuração entradas saídas e interrupção
  - `voidSetup(){`  
    `attachInterrupt(0, irVermelho,FALLING);`  
    ...  
    `}`
- 3º PASSO: Sub rotinas do programa:
  - `voidirVermelho(){`  
    ...  
    `}`
- 4º PASSO: Lógica do programa (`voidLoop()`)
- 5º PASSO: Compilar e enviar para o Arduino





# Projeto 2 - Semáforo

```
int passo = 1;
```

→ Variável global para armazenar o passo atual.

```
int vermelho = 12, amarelo = 11, verde = 10;
```

→ Declaração dos LEDs usados conforme o circuito.

```
int botao = 2;
```

→ Declaração do botão conforme o circuito.

```
void setup(){
```

→ Aqui temos a função de configuração.

```
  attachInterrupt(0, irVermelho, FALLING);
```

→ Vinculamos a função `irVermelho()` com uma interrupção externa na borda de descida do canal 0 ( no pino 2).

```
  pinMode(botao, INPUT);
```

→ Configuramos o pino do botão como entrada.

```
  pinMode(verde, OUTPUT);
```

→ Configuramos os pinos dos LEDs como saídas digitais.

```
  pinMode(amarelo, OUTPUT);
```

```
  pinMode(vermelho, OUTPUT);
```

```
}
```



# Projeto 2 - Semáforo

```
void irVermelho(){  
  p1();  
  passo = 1;  
  return;  
}
```

→ Essa função liga o LED vermelho e desliga os outros, e retorna o passo para o primeiro\*.

```
void p3(){  
  digitalWrite(vermelho,LOW);  
  digitalWrite(amarelo,HIGH);  
  digitalWrite(verde,LOW);  
  return;  
}
```

→ Função para colocar os LEDs no passo 3.

```
void p2(){  
  digitalWrite(vermelho,LOW);  
  digitalWrite(amarelo,LOW);  
  digitalWrite(verde,HIGH);  
  return;  
}
```

→ Função para colocar os LEDs no passo 2.

```
void p1(){  
  digitalWrite(vermelho,HIGH);  
  digitalWrite(amarelo,LOW);  
  digitalWrite(verde,LOW);  
  return;  
}
```

→ Função para colocar os LEDs no passo 1.



# Projeto 2 - Semáforo

```
void loop(){  
  
  if(passo == 1){  
    p1();  
    passo = 2;  
    delay(5000);  
  }  
  
  if(passo == 2){  
    p2();  
    passo = 3;  
    delay(5000);  
  }  
  
  if(passo == 3){  
    p3();  
    passo = 1;  
    delay(1000);  
  }  
}
```

→ Função loop – aquela executada sempre que o microcontrolador estiver alimentado.

→ Aqui caso a variável que guarda o passo atual estiver em 1, executamos a função que coloca os LEDs no passo 1, então mudamos a variável passo para indicar o novo passo e esperamos 5.000ms ou 5s.

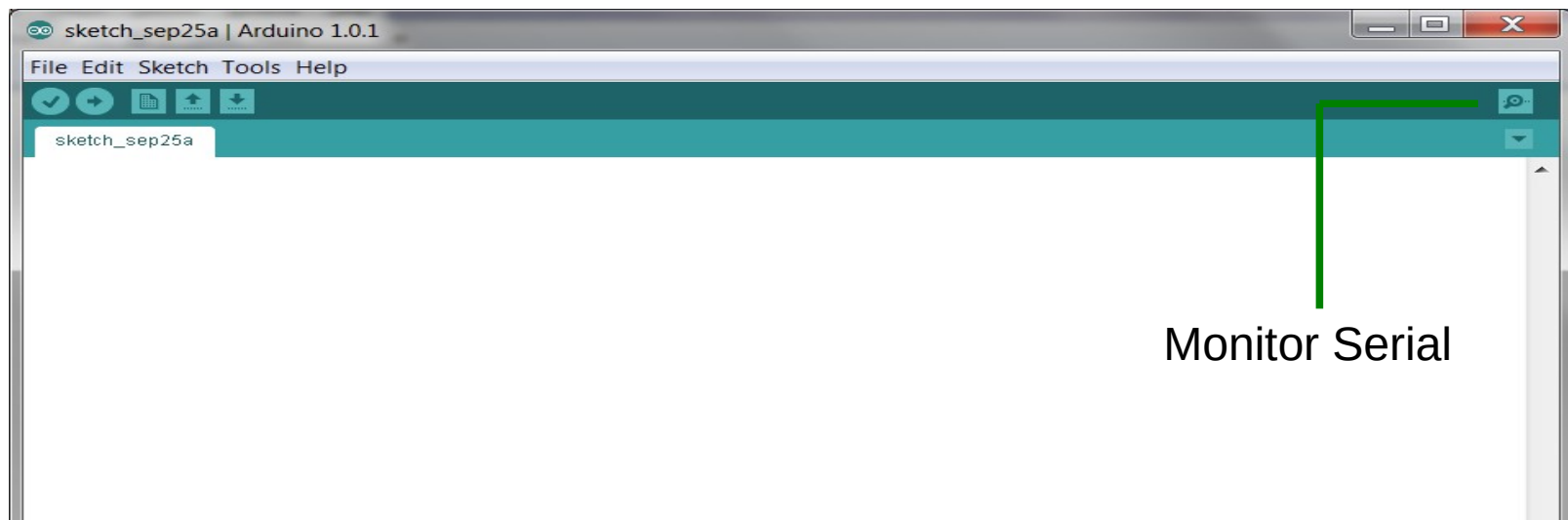
→ Faz o mesmo que o anterior porem para o segundo passo.

→ Executa o mesmo para o terceiro passo, porem agora esperando apenas 1 segundo no sinal amarelo.



# Comunicação Serial

- Faz “debug” dos programas
- Envia e recebe dados no formato serial







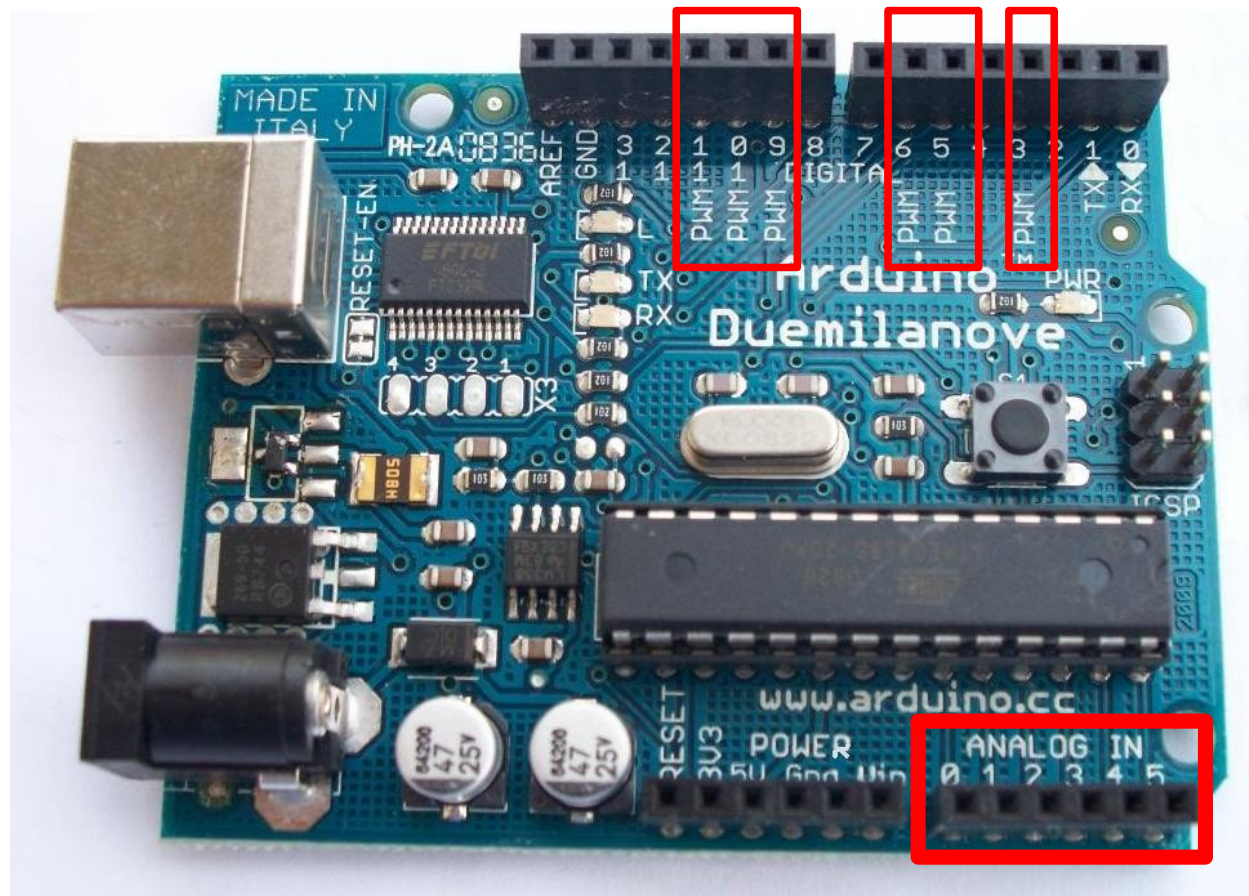
# Comunicação Serial

Funções:

- `Serial.begin(taxa)`
- `Serial.print()`, `Serial.println()` e `Serial.write()`
- `Serial.read()` e `Serial.peek()`
- `Serial.available()`



# Entradas e Saídas Analógicas

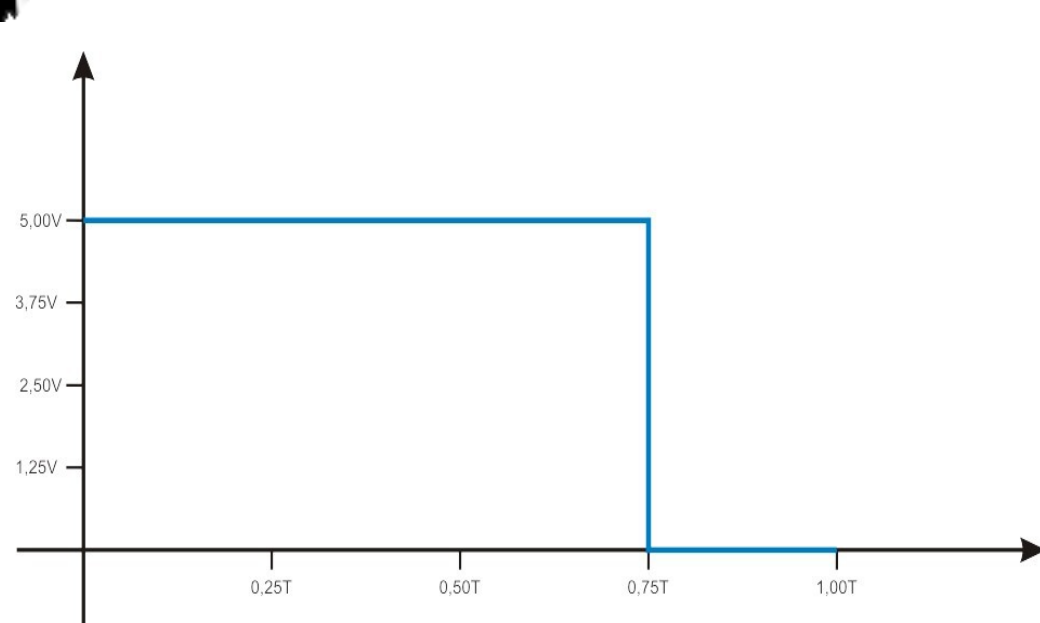


Fonte: <http://www.reuk.co.uk/wordpress/standard-alone-arduino-on-a-breadboard/>

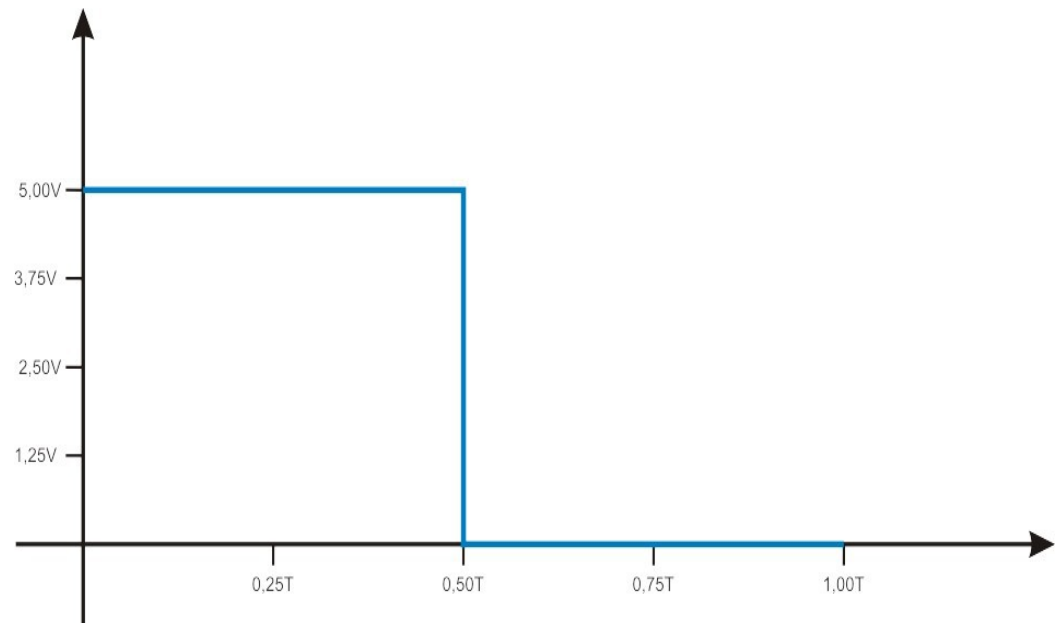


# Saídas Analógicas

- Saídas no Arduino usam PWM.



Ciclo ativo de 75%



Ciclo ativo de 50%



# Saídas Analógicas

`analogWrite`(pino, valor) , onde:

- pino : pino de saída do PWM.
- valor : valor entre 0 ( 0%) e 255 ( 100%) que determina a porcentagem em que o sinal estará ativo.

Obs.: Não é preciso declarar o pino como saída(usando `pinMode()`) para usá-lo PWM.





# Entradas Analógicas

- Lê valores de Tensão de 0V~5V
- Precisão de 10bits
- Sendo 0 para 0v e 1023 para 5v



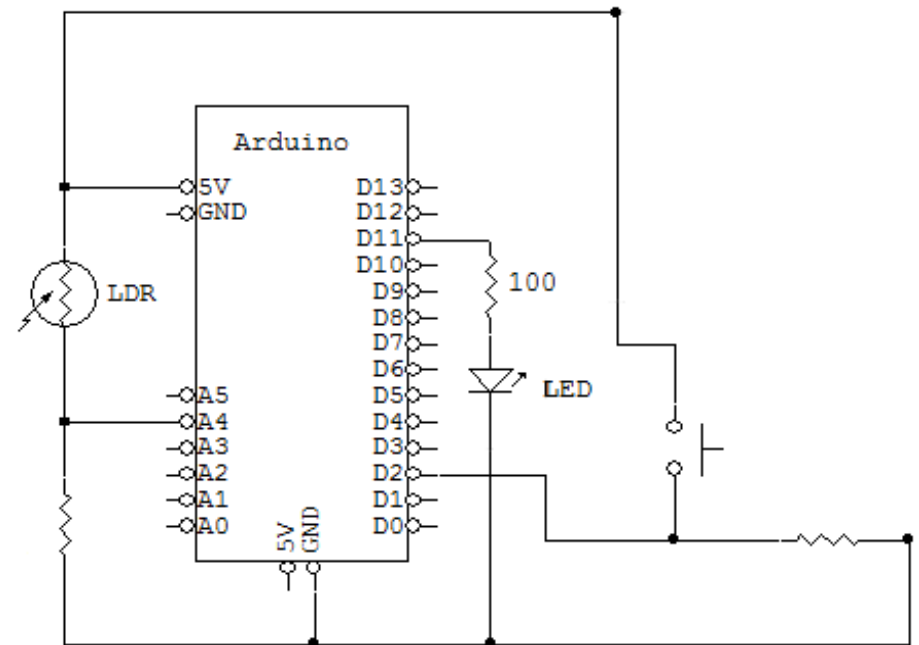
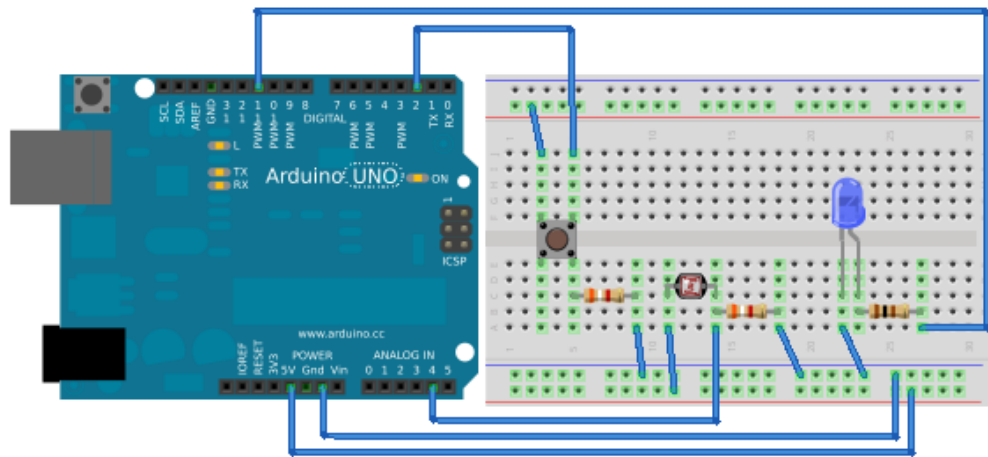
# Entradas Analógicas

- `analogRead(pino)` , onde “pino” indica o pino o qual será feito a leitura.
- Essa função retorna um valor inteiro de 0 a 1023 correspondente a leitura feita.



# Projeto 3 – Controle de Luminosidade

## Hardware e esquemático



Sketch feito com Fritzing



# Projeto 3 – Controle de Luminosidade

## FUNÇÕES IMPORTANTES

- **map**(valor, minIn, maxIn, minOut, maxOut)
  - valor : é a variável que contém o valor que se deseja adaptar.
  - minIn : valor mínimo da entrada.
  - maxIn : valor máximo da entrada.
  - minOut: valor mínimo da saída
  - MaxOut: valor máximo da saída
- **boolean** variável = **status** (**true** ou **false**)
- **Serial.begin**(valor) e **Serial.println**(variável)



# Projeto 3 – Controle de Luminosidade

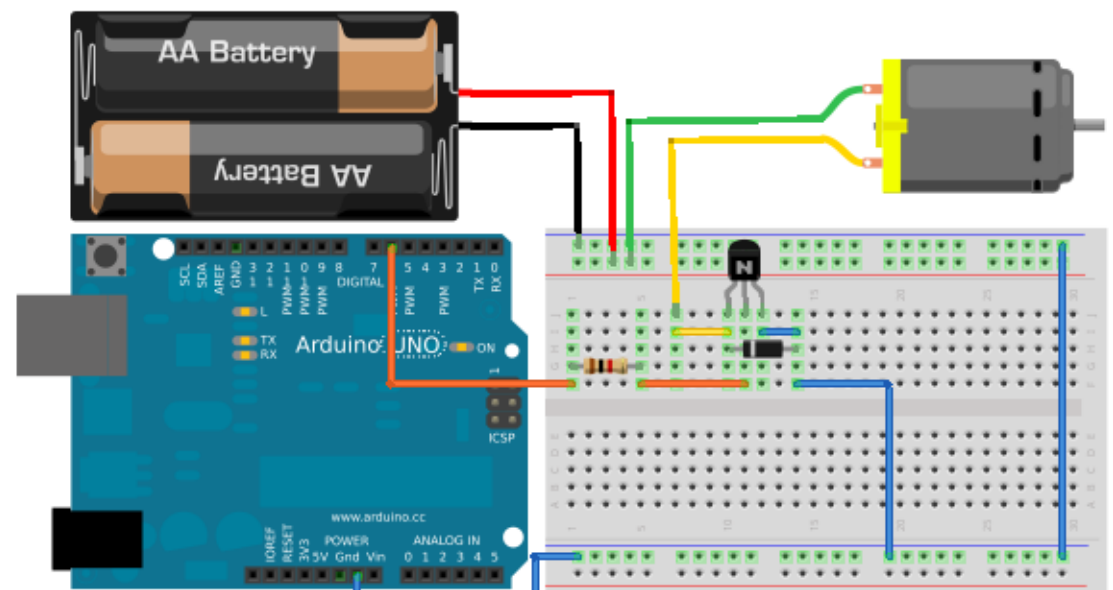
- 1º PASSO : Descobrir a faixa de valores lidos pelo LDR usando a comunicação serial.
- 2º PASSO: Ajustar os valores lidos para escrevê-los na saída correspondente à do LED
- 3º PASSO: Fazer a rotina do programa





# Motores DC

- Corrente máxima por porta : 40mA
- Necessidade de um circuito auxiliar
- Transistor BC458 ou TIP122 (até 5A)

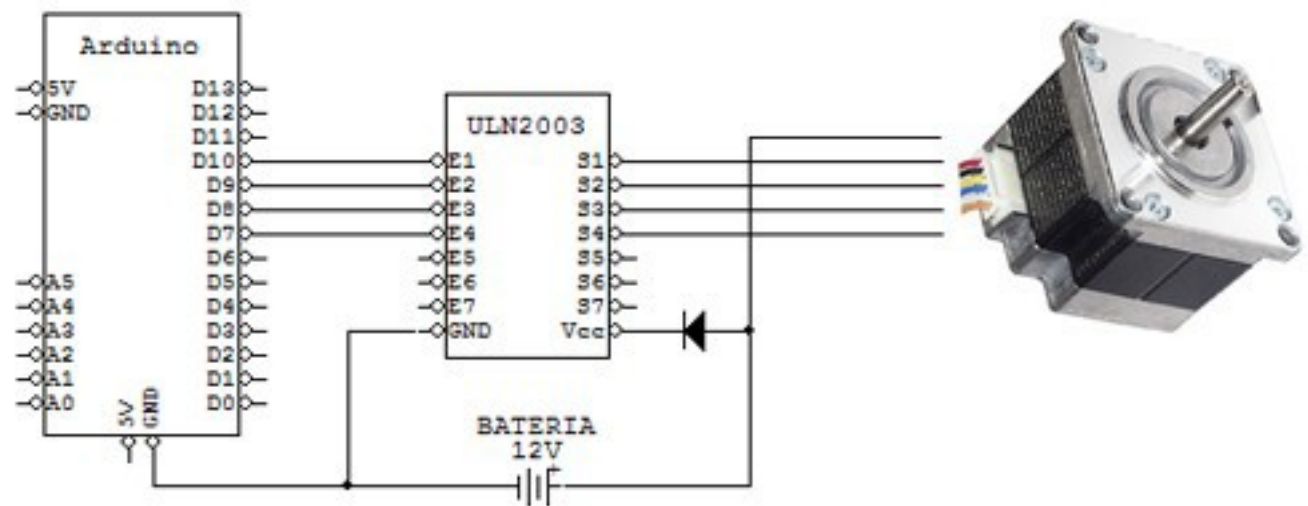


Sketch feito com Fritzing



# Motores de Passo

- Girar quantidade específica de graus
  - Imã Permanente ( $7,5^\circ$  a  $15^\circ$ )
  - Híbrido ( $3,6^\circ$  a  $0,9^\circ$ )
- Necessidade de um CI ULN2003 (entradas barradas)



Sketch feito com Fritzing



# Projeto Novo: DHT11 Sensor Temperatura

O sensor da linha DHT\* faz o sensoreamento de temperatura e umidade através de uma saída digital calibrado.

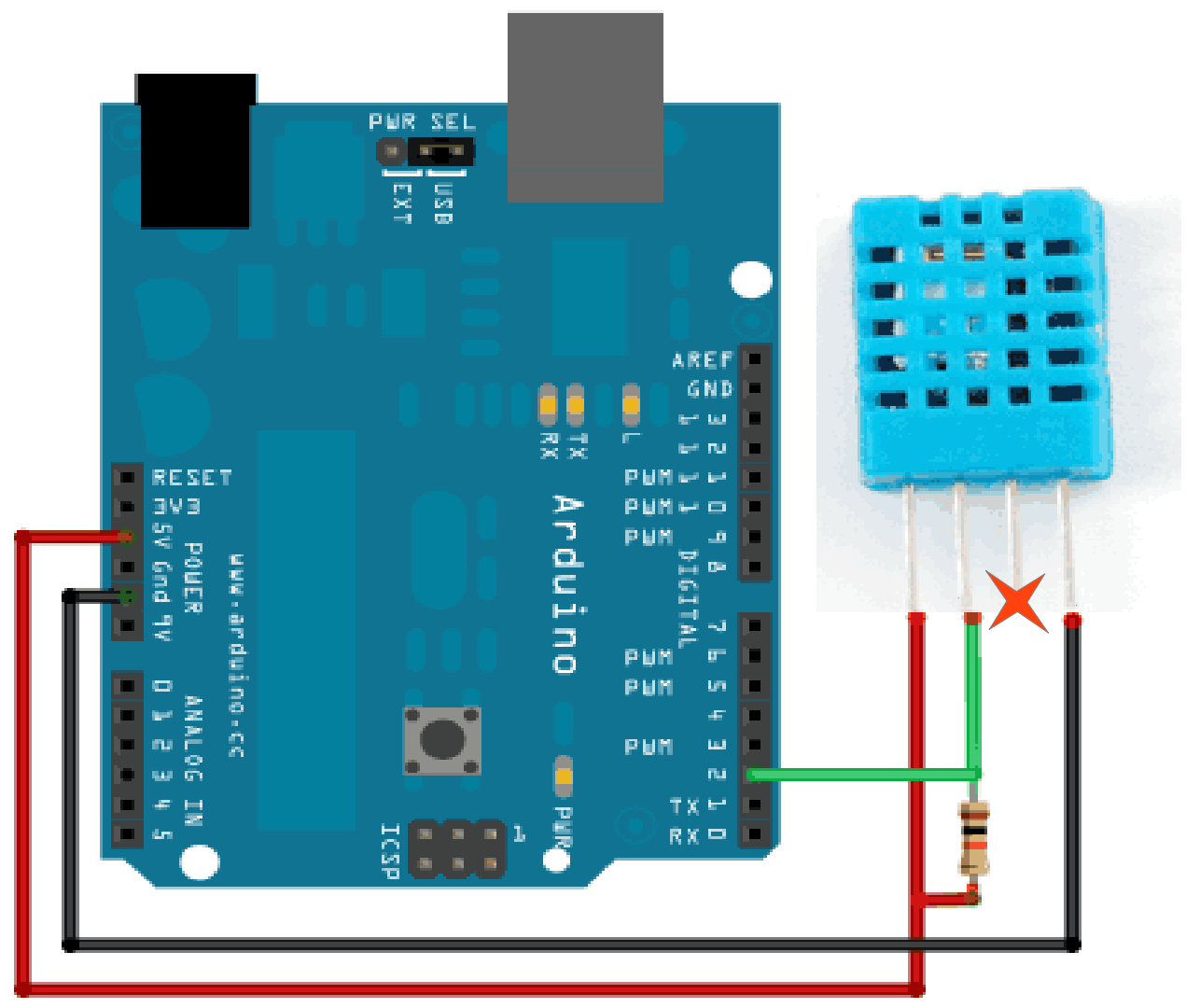
O sistema de interface é feito no modo one wire, tornando a ligação mais simples.

- Temperatura de 0 a 50°C, erro de +/- 2°C,
- Umidade: 20 a 90% RH, erro de +/- 5%;

[Fonte texto: <http://www.micro4you.com/files/sensor/DHT11.pdf>]



# Projeto Novo





# Projeto Novo LCD 16x2

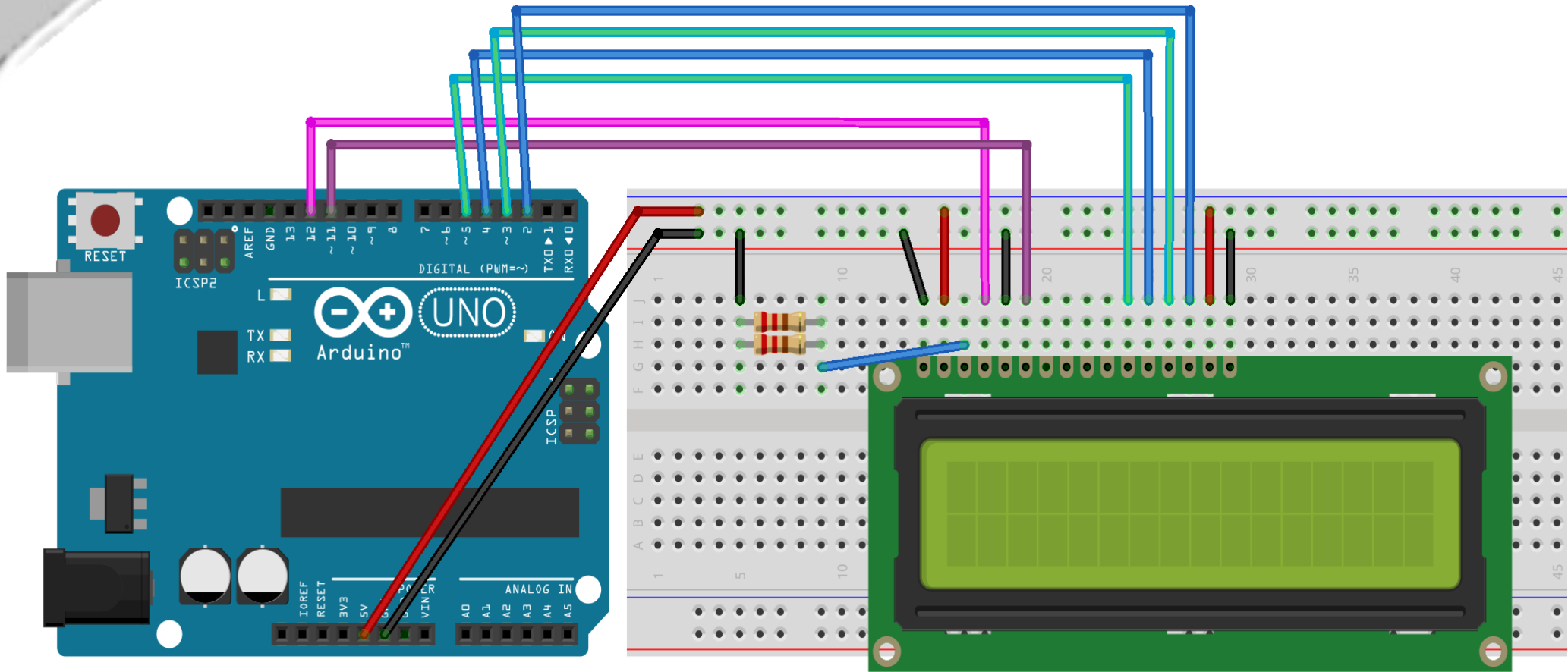
Um diferencial em qualquer projeto microcontrolado é utilização de um display LCD para indicação de parâmetros e informações diversas.

O modelo utilizado neste material é o JHD 162A, possui backlight com ajuste de luminosidade e contraste dos caracteres controláveis. É fácil de encontrá-lo no mercado por ter o preço mais acessível.





# Projeto Novo

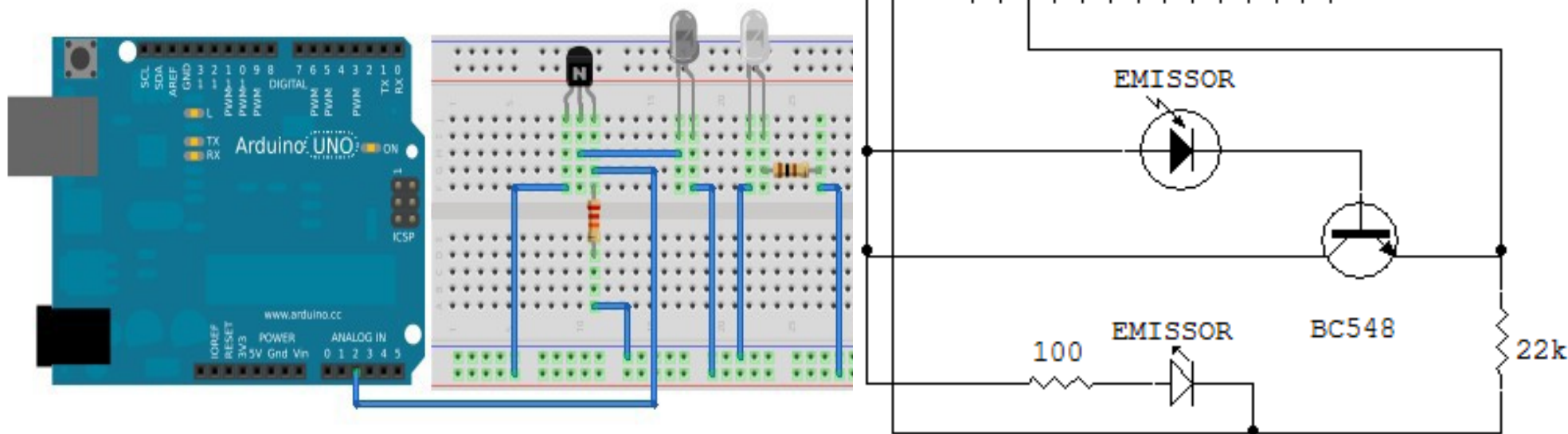


Sketch feito com Fritzing



# Projeto 4 – Verificação de Superfície

Objetivo: Distinguir a cor branca da cor preta em uma superfície



Sketch feito com Fritzing



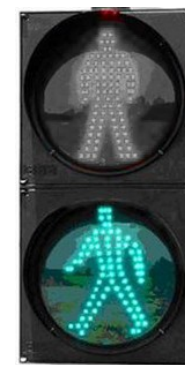
# Projetos Extras

- Projeto Extra 1 - Semáforo
- Projeto Extra 2 – Controle de Luminosidade



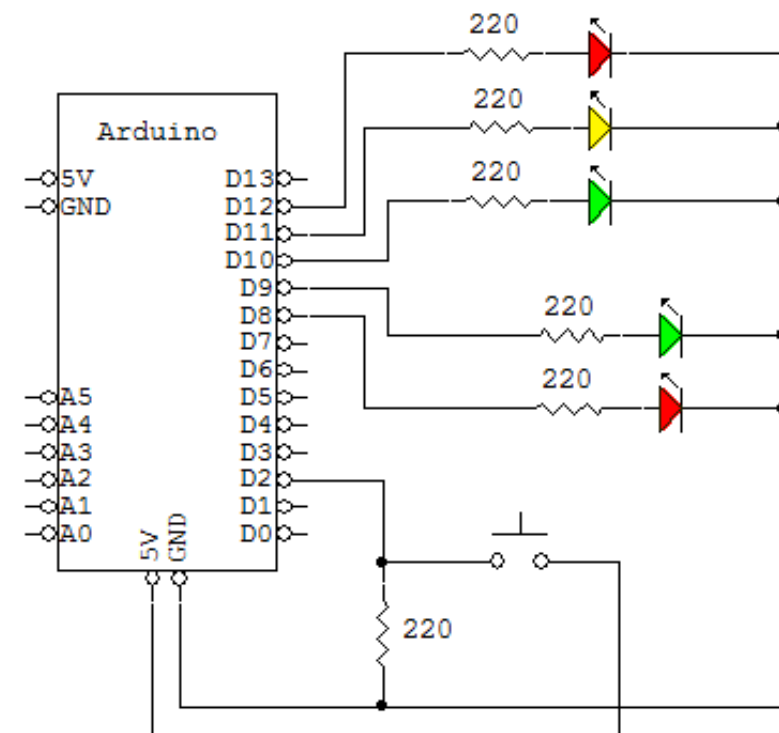
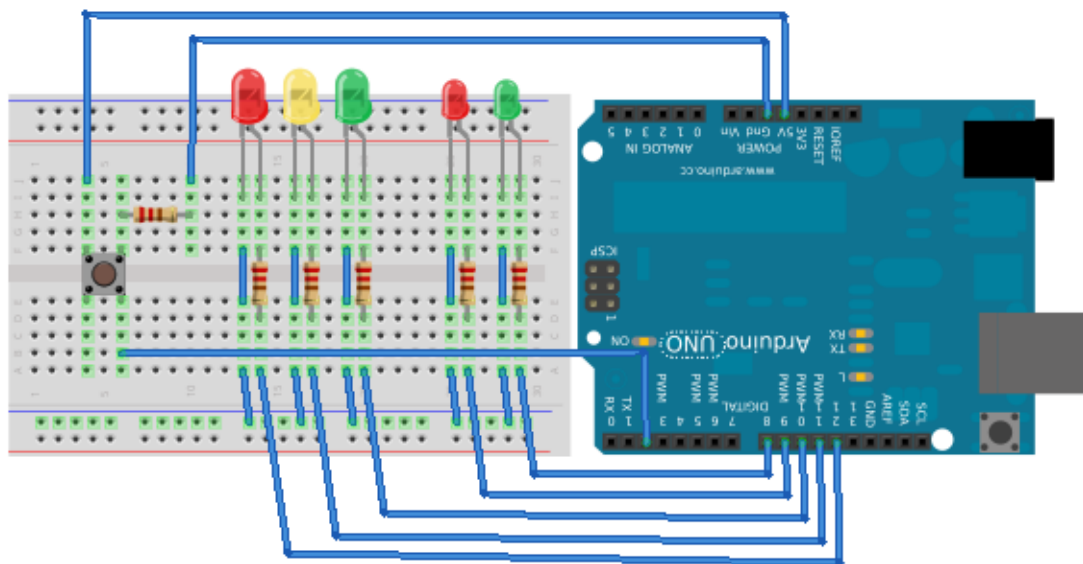
# Projeto Extra 1 - Semáforo

Objetivo: Projetar um semáforo mais próximo do que o encontrado nas ruas com controle de pedestres.





# Projeto Extra 1 - Semáforo



Sketch feito com Fritzing





# Projeto Extra 1 - Semáforo

- 1º PASSO: Declarar variáveis
- 2º PASSO: Configuração entradas saídas e interrupção
  - `voidSetup(){`  
    `attachInterrupt(0, reduzTempo,FALLING);`  
    `}`
- 3º PASSO: Sub rotinas do programa:
  - `void reduzTempo){`  
    `if(tempo>=10&& tempo<=19)tempo=19;`  
    `}`  
    ...
- 4º PASSO: Lógica do programa (`voidLoop()`)
- 5º PASSO: Compilar e enviar para o Arduino



# Oficina: Objetivo

- Integrar o Arduino a uma rede de computadores.
- Para esta oficina nossa planta será uma sala com temperatura, luminosidade e acesso controlados.

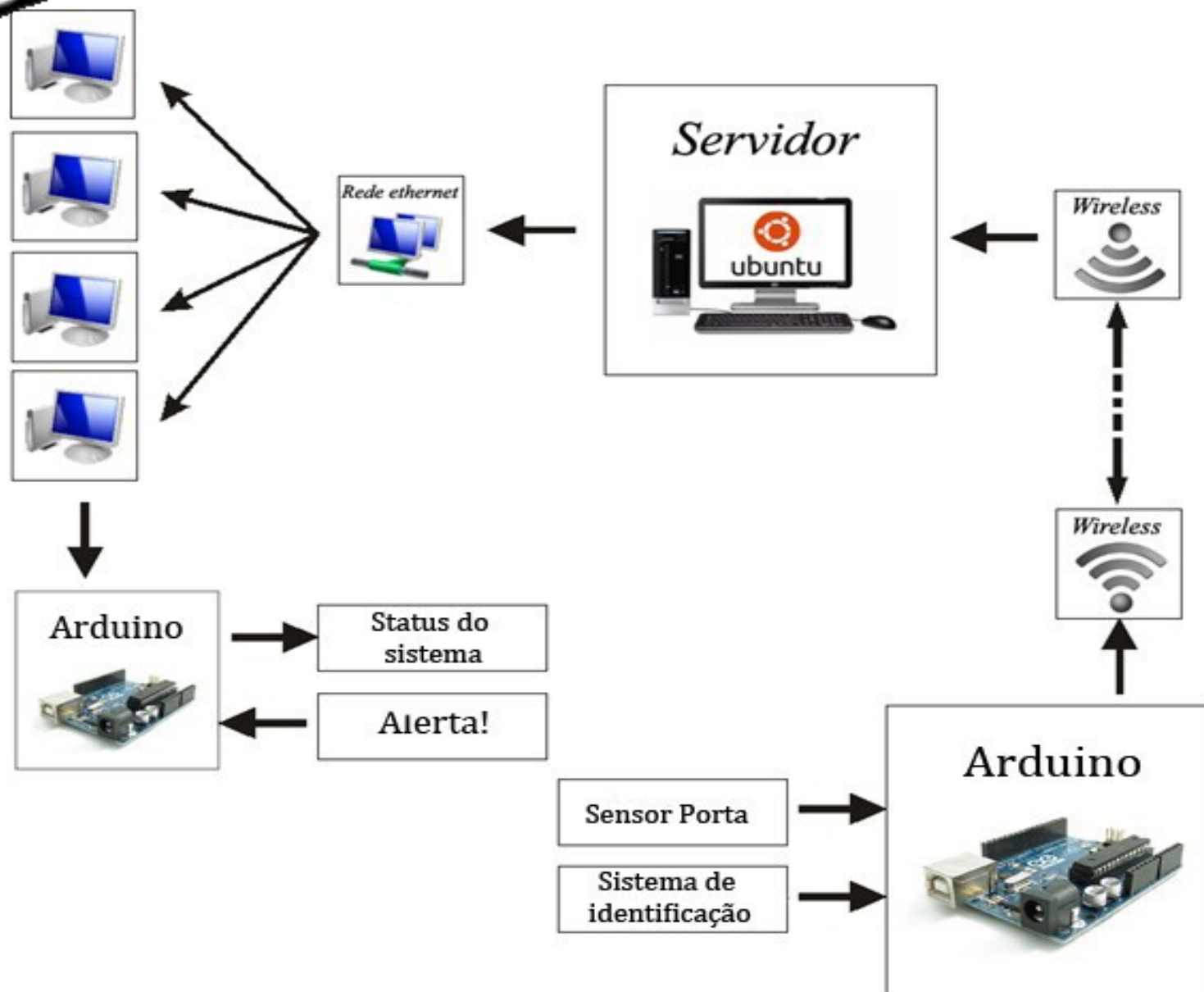


# Oficina: Contexto

- Para manter a temperatura controlada a porta do local deve permanecer fechada, e quando aberta deve ficar aberta pelo mínimo de tempo possível.
- Ao entrar pela porta cada pessoa deve identificar-se.
  - O sistema deve guardar uma lista das pessoas que entram no local.



# Oficina





# Bibliografia e Materiais para consulta

- <http://www.arduino.cc>
- <http://vimeo.com/31389230>
- <http://fritzing.org/>

[Main page do Arduino]

[Documentário Arduino]

[Software para desenho de circuitos]





# Perguntas?

- Site: [www.colmeia.udesc.br](http://www.colmeia.udesc.br)
- E-mail: [contato@colmeia.udesc.br](mailto:contato@colmeia.udesc.br)

Obrigado!