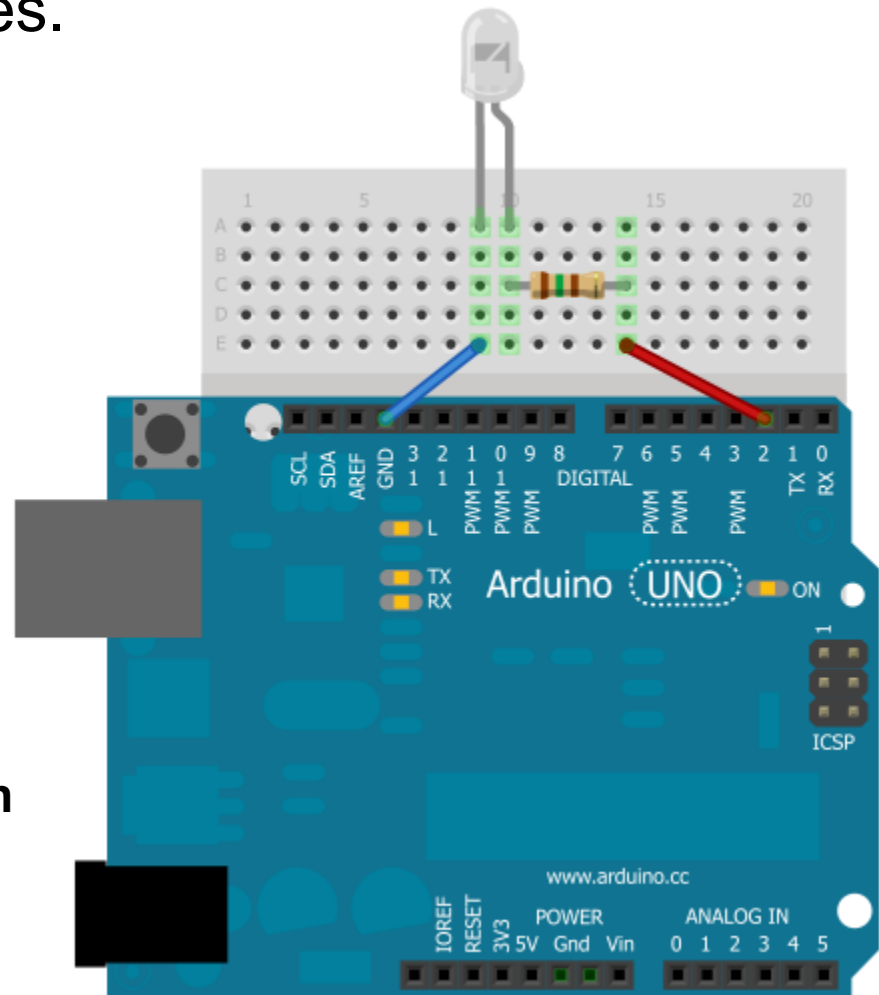


Robótica com Arduino

Hoje iremos conectar LEDs externos ao Arduino e fazer exercícios com eles.



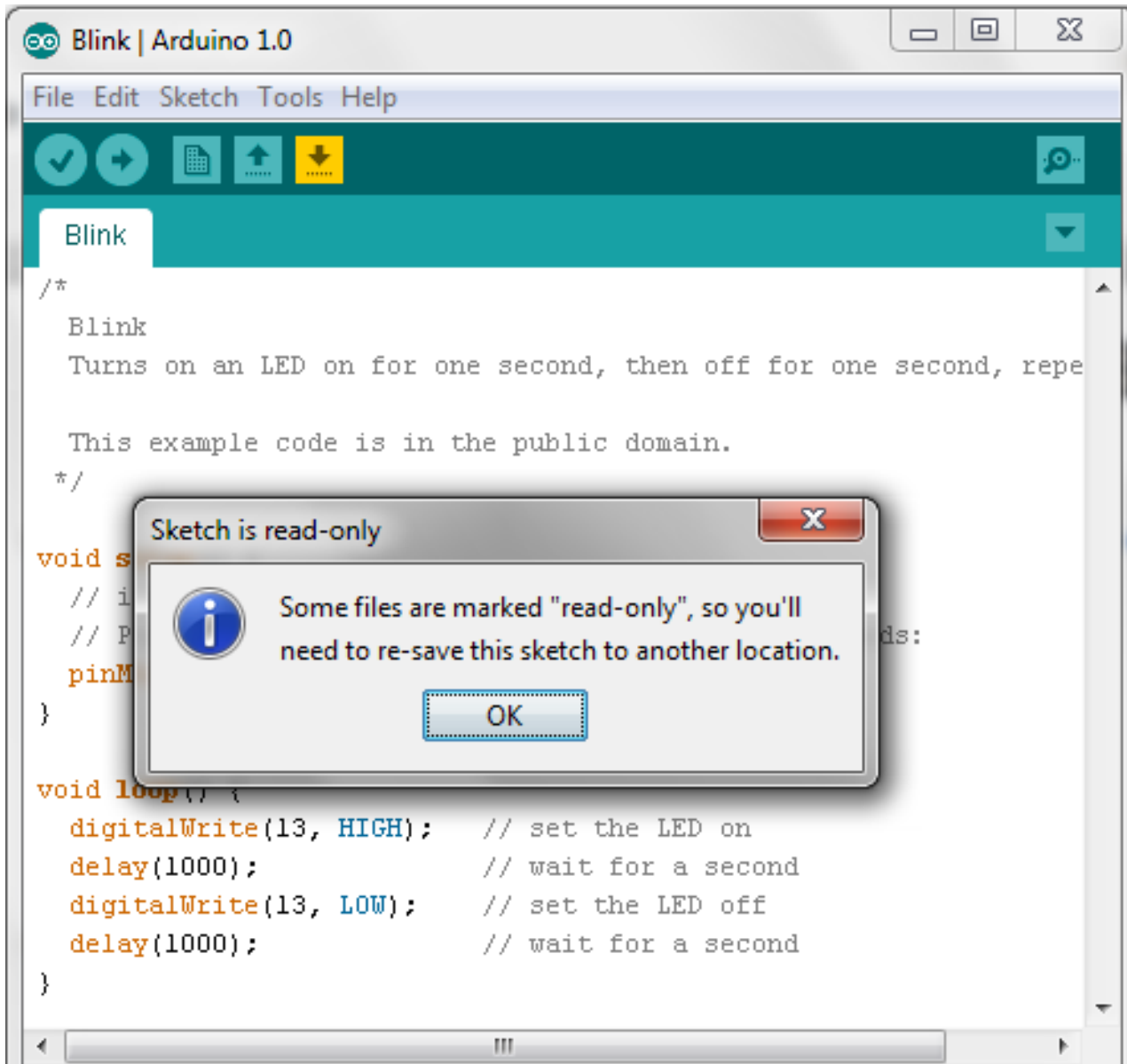
Professor: Paulo Marcos Trentin

Salvando um projeto na IDE do Arduino

Será cada vez mais comum, precisarmos criar novos programas no Arduino e salvá-los separadamente.

A **IDE** (Integrated Development Environment - Ambiente Integrado de Desenvolvimento) do Arduino permite-nos isso.

Abra o programa de exemplo Blink e tente salvá-lo



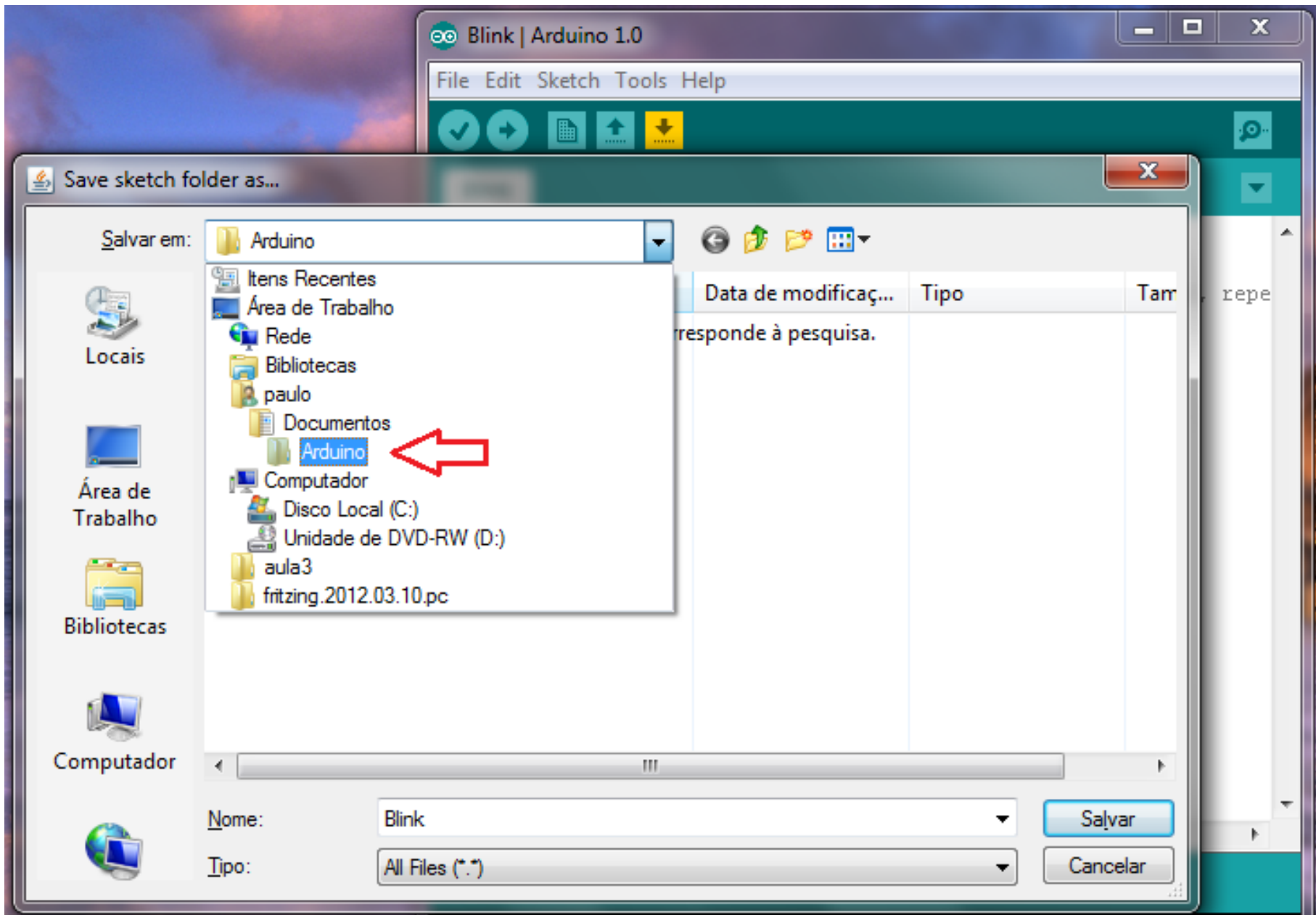
Mensagem de Erro ao salvar programa Blink

A mensagem vista, informa que o arquivo que você está tentando salvar não pode ser sobrescrito (pois é somente de leitura).

Ao pressionar **Ok**, ele pede para você salvar o arquivo no seu "livro de rascunhos", ou o **sketch book**.

Ao confirmar, ele salvará então o exemplo Blink, dentro da pasta **Documentos** do windows

Salvando sketch - rascunho no Arduino



Abrindo sketch - rascunho no Arduino

Esta pasta [Arduino](#), dentro de [Documentos](#), conterá todos os seus rascunhos, ou seja, todos os programas que você cria com a IDE.

Uma vez salvo, você pode editar seu programa normalmente e ir salvando a cada mudança feita nele.

Para recuperar um programa já salvo, basta acessar o menu [File](#) -> [Sketchbook](#) e escolher o programa na lista

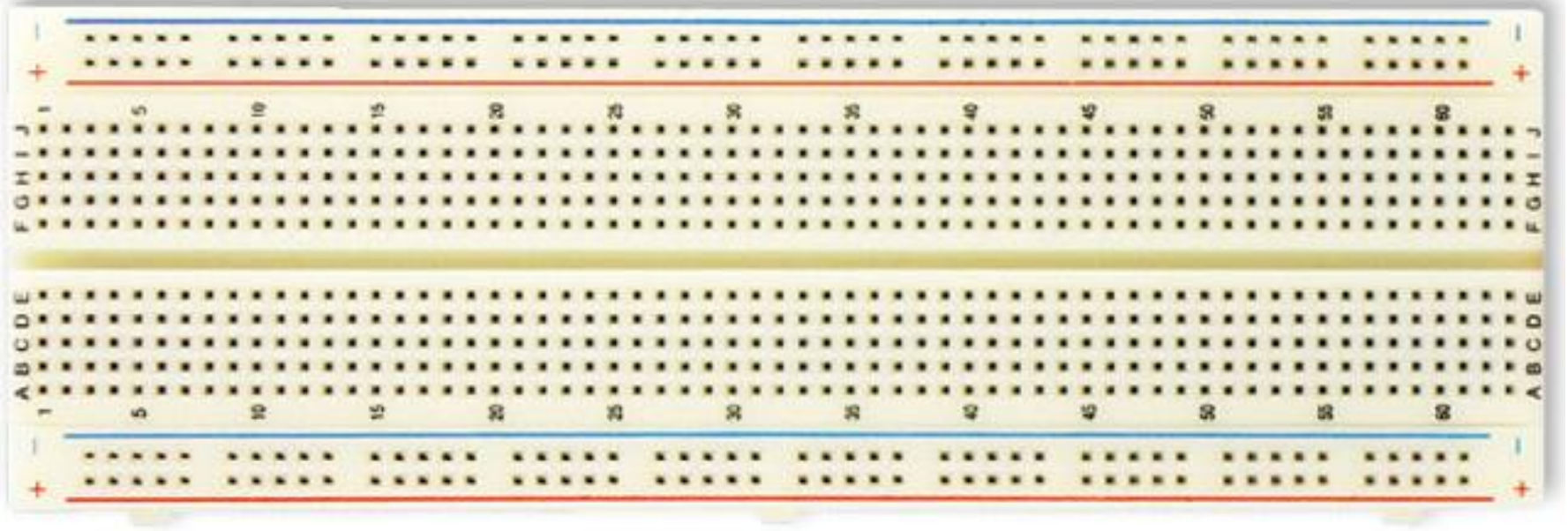
File Edit Sketch Tools Help

New	Ctrl+N
Open...	Ctrl+O
Sketchbook	
Examples	
Close	Ctrl+W
Save	Ctrl+S
Save As...	Ctrl+Shift+S
Upload	Ctrl+U
Upload Using Programmer	Ctrl+Shift+U
Page Setup	Ctrl+Shift+P
Print	Ctrl+P
Preferences	Ctrl+Comma
Quit	Ctrl+Q

Blink
Exercicio1
Exercicio2
Meu_programa

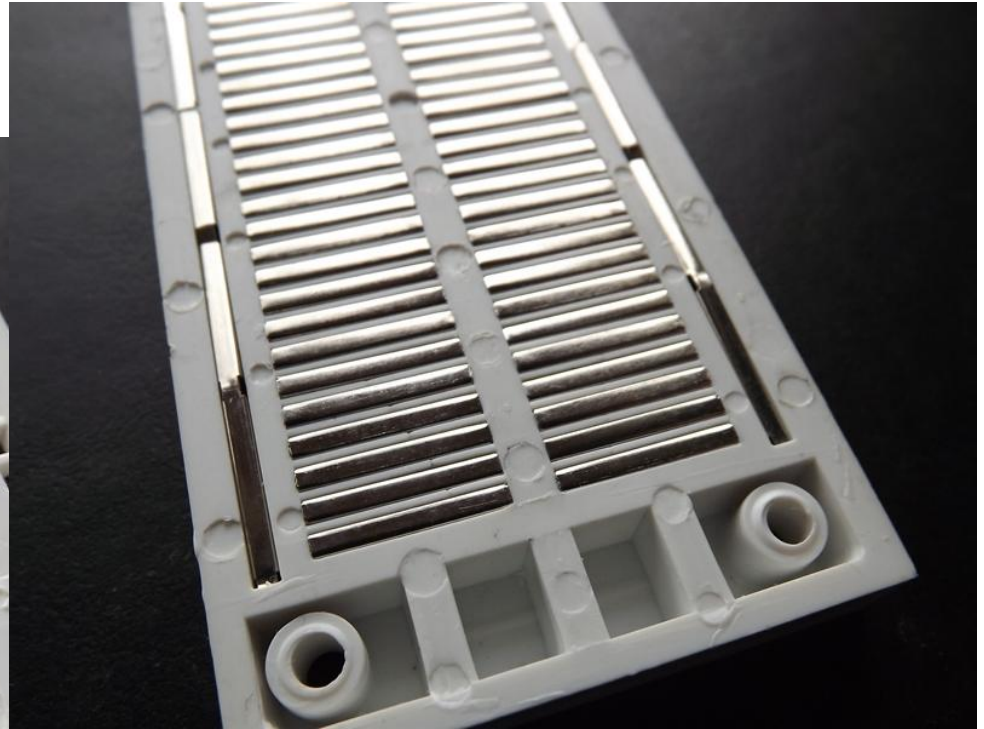
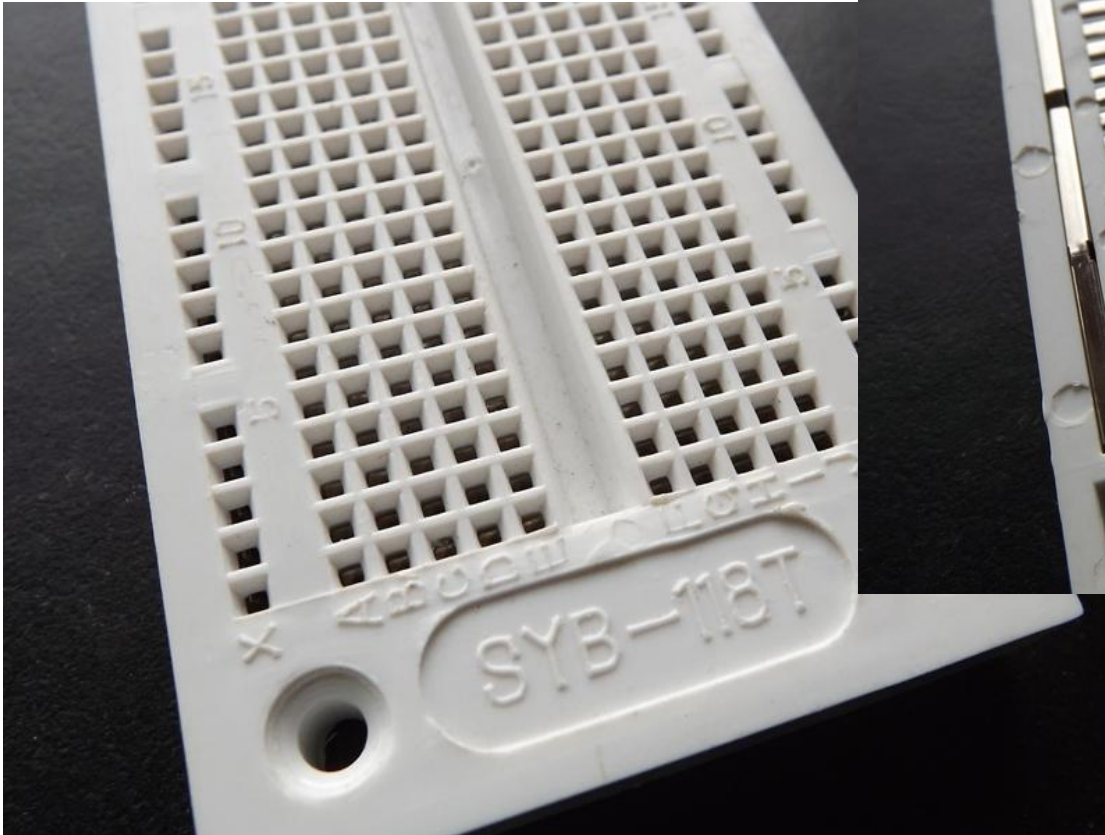
Conectando LEDs na Protoboard

A placa de prototipação ajuda-nos a fazer muitos testes de forma rápida e sem solda!



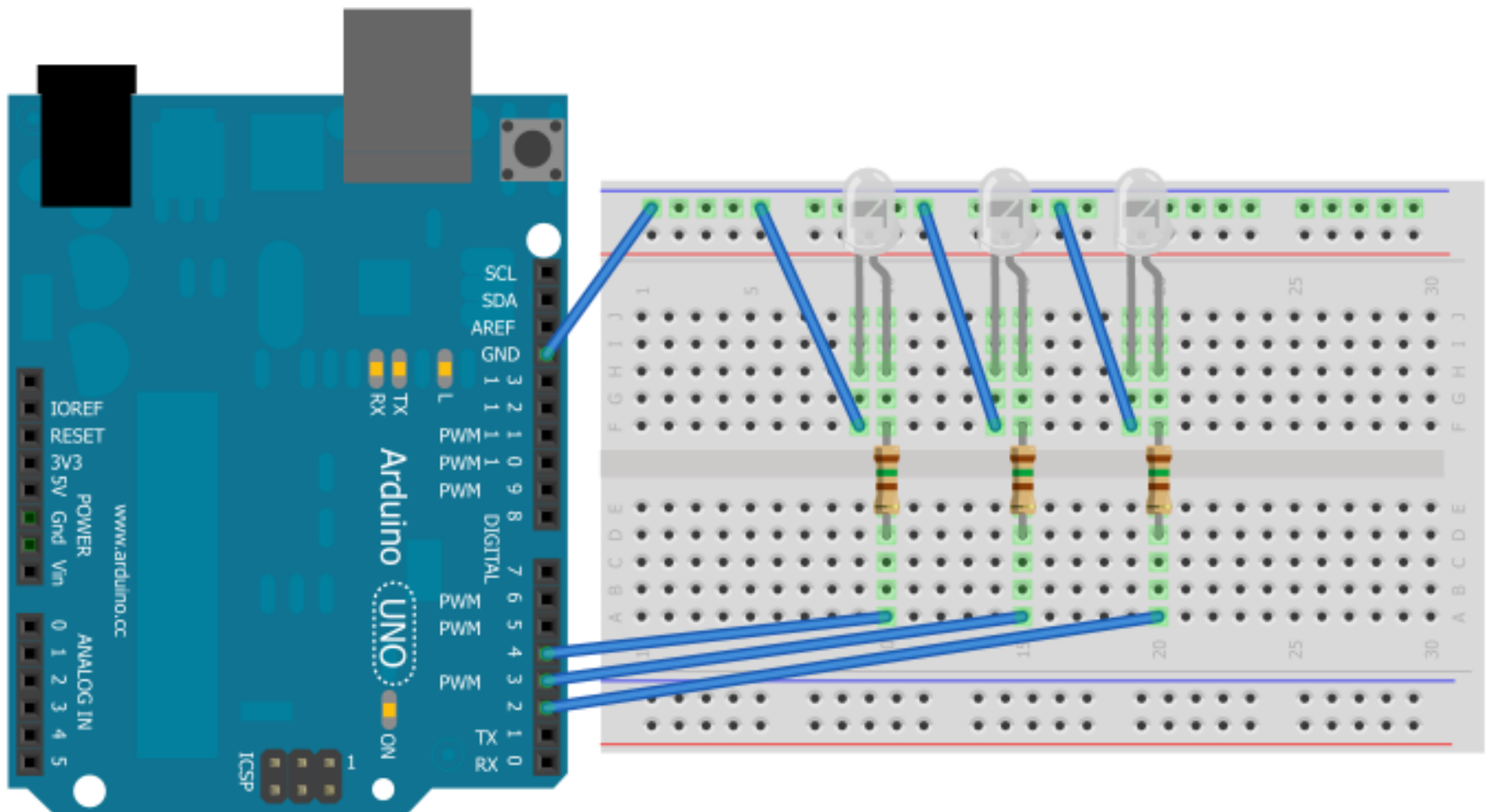
Funcionamento da Protoboard

Sem muitos mistérios, ela funciona com colunas e linhas. Veja a foto abaixo para mais detalhes



Exercício 1

Conecte 3 LEDs e resistores na protoboard, seguindo o desenho abaixo.



Exercício 2

Faça um programa para ligar e desligar os 3 LEDs de forma sequencial. Conecte os LEDs nas portas digitais 2, 3 e 4 do Arduino.

Primeiro deve-se ligar o 1º LED, desligá-lo e só então ligar o 2º LED e assim por diante.

O intervalo de acionamento deve ser de 200 milissegundos

Exercício 3

Crie um novo programa para que ligue os 3 LEDs com intervalo de 300 milisegundos entre eles.

Após todos estarem ligados, então o programa deve desligar cada LED, também com intervalo de 300 milisegundos.

Exercício 4

Altere as portas dos LEDs para as portas digitais 5, 6 e 7 do Arduino, e o tempo entre os intervalos para 500 milissegundos.

Lembre-se de alterar as portas no seu software também!

Exercício 4 - continuação

Pense agora, como evitar tanto trabalho?





O pulo do gato

E se em vez de digitarmos sempre o número do tempo de atraso, nós simplesmente o substituirmos por uma variável?

`delay(500);` ficará assim: `delay(tempoAtraso);`

O que é uma variável?

Imagine-a como uma caixa de papel.
Dentro dela cabe algo, pode ser "qualquer coisa", desde que não grande demais.

Ela pode estar vazia ou cheia

Podemos trocar seu conteúdo



Criando uma variável no Arduino

```
int tempoAtraso = 500;
```

int -> informa que essa variável armazena somente números inteiros de -32.768 até 32.767

tempoAtraso -> é o nome da nossa caixa, pode ser qualquer nome desde que comece com uma letra e não tenha caractere especial.

Onde crio uma variável?

Se for usá-la em todo o programa, insira-a antes de tudo, fora das funções. Ela então será uma variável Global. Lembre-se deste nome.

```
// Cria variável global contendo o tempo de atraso
```

```
int tempoAtraso = 500;
```

```
void setup() {
```

```
    // define o pino 2 como sendo de saída
```

```
    pinMode(2, OUTPUT);
```

```
    ....
```

```
}
```

```
void loop() {
```

```
    // Envia 5 volts para o pino 2 do Arduino
```

```
    digitalWrite(2, HIGH);
```

```
    // Aguarda meio segundo (500 ms)
```

```
    delay(tempoAtraso);
```

```
    ....
```

Por que **int**?

int é um identificador (pg 3 da apostila). Ele informa ao compilador que a variável em questão irá usar 2 bytes (16 bits!) de memória RAM para armazenar seu valor.

$2^1 = 2$ possibilidades (com 1 bit, podemos ter apenas dois estados)

$2^{16} = 65536$ possibilidades. Ou seja, podemos armazenar qualquer valor entre 0 e 65535. Ou qualquer valor entre -32768 até +32767. ($65536 / 2 = 32768$. 32767 pois o **0** também conta)

Exercício 5

Altere o programa do Exercício 4 que liga os LEDs de forma sequencial no tempo de 500 milissegundos, inserindo nele variáveis no lugar do número dos LEDs e no lugar do tempo de parada.

Esse código fonte ficou melhor que o outro? Por que?

Trabalho avaliativo 1

Criar 5 programas, um para cada exercício feito aqui;

- Salvar cada programa com seu nome. Ex.: Exercicio1, Exercicio2 na Sketchbook;
- Todos os exercícios devem ter comentários claros do que cada parte do programa faz;
- Dentro da Sketchbook, criar um arquivo de texto respondendo a pergunta do exercício 5;
- Ainda neste arquivo de texto, coloque o nome completo de cada membro da dupla ou trio, seguido por seu email.