

Pretende-se que realizes na prática a atividade do LED a piscar realizada em prática simulada no Tinkercad no guião anterior. Para isso irás necessitar do seguinte material:



No teu diário de bordo, cria um novo título no final do documento de nome Arduino Laboratório 2 – Piscar 1 LED.

Começa por **montar** os **seguintes componentes** na **Breadboard**, tal como mostra a **figura seguinte**:



Estabelece as ligações corretamente com o teu Arduino por forma a utilizar a saída digital 8 para controlar o Led.

Informática

Arduino – Led a piscar

Laboratório Nº2



Abre o IDE do Arduino.

sketch_oct28	= Arduino IDE 2.2.1 =	- 0	×
00	Selacionar Placa •	4	۰Q-
sketch,	oct28a.ino		
1	void setup() {		
	// put your setup code here, to run once:		
p-0. 5			
6	<pre>void loop() {</pre>		
7	// put your main code here, to run repeatedly:		
8			
9			
10			
	to 4 Math. Million Review and		

Adiciona o seguinte código ao teu IDE para poderes fazer piscar o teu LED: Declaração dos pinos a utilizar como constantes:

#define ledPin 8

Ou, em **alternativa**, podes definir os **pinos** de outra forma:

const int ledPin =8;

Repara que as **variáveis** são **definidas** com **letra grande** sempre que **muda** a **palavra** no **nome** da variável. Isto é uma **técnica** utilizada pelos **programadores** para **melhor ler** o **nome** de **variáveis**.

Depois de **definirmos** os **pinos** temos de **indicar** se são **vão** ser de **saída** ou **entrada**. Como **queremos acender Leds** queremos **escrever** no **pino** logo **são** de **saída** (OUTPUT). Para isso na função *setup(),* a que só é **lida uma vez** ao **iniciar** o **programa dizemos** o que cada um dos **pinos definidos será**:

```
void setup() {
    pinMode(ledPin, OUTPUT);//função que define se aporta digital é de output ou input
}
```

Finalmente o programa em si. Na função *loop()* será escrito o programa que irá correr para sempre em repetição até que o Arduino seja desligado. Neste caso queremos ligar o LED verde 4 segundos.

```
void loop() {
   digitalWrite(ledPin, HIGH); //escrever no pino a alto 5V
   delay(4000); // Espera 4000 milisegundos ou 4s
   digitalWrite(ledPin, LOW); //escrever no pino a baixo 0V
   delay(4000); // Espera 4000 milisegundos ou 4s
}
```

Falta enviar o programa para o Arduino. Relembro os passos para isso acontecer. Selecionar o Arduino com que vais trabalhar:

sketch_oct28a Arduino IDE 2.2.1								Arduino Yún
Arquivo	Editar R	lascunho	Ferramentas Ajuda					Arduino Uno
Ø	€ €	Sele	Autoformatação Arquivar Esboco	Ctrl+T				Arduino Uno Mini Arduino Duemilanove or Diecimila
Ph	sketch_o	ct28a.ino	Gerenciar bibliotecas	Ctrl+Shift+I				Arduino Nano
	1	void s	Monitor Serial	Ctrl+Shift+M				Arduino Mega or Mega 2560
53	2	// p	Serial Plotter			2:		Arduino Mega ADK
	3	}			-			Arduino Leonardo
D-D-	5	·	Firmware Updater					Arduino Leonardo ETH
ШИ	6	void l	Enviar certificado Raiz SSL					Arduino Micro
	7	// p	Placa		۲	Gerenciador de Placas Ctrl+Shift+B		Arduino Esplora
₩>	9	}	Porta		Þ	Andrian AVO Decede		Arduino Mini
	10	·	Obter Informações da Placa			Arquino AVK Boards	P	Arduino Ethernet
Q		G	Gravar bootloader					Arduino Fio Arduino BT







Eiga o Arduino à porta USB do teu PC e seleciona a porta COM a que está ligado. Exemplo:



De seguida verifica o código para ver se não há erros:



Caso não tenha erros clica agora no botão para enviar o programa para o Arduino.



Faz um **pequeno vídeo** com o **telemóvel** do **circuito + Arduino com o LED a piscar** e envia para o teu **Drive** para a **pasta Arduino - Vídeos**. **Altera** o nome do **ficheiro** de vídeo **para 2** – **Led a piscar**.

Service Tira também uma **foto** e **cola** no teu **diário** de **bordo**.

Guarda o programa, no IDE do Arduino com o nome 2LEDpiscar. Envia este ficheiro para o teu drive para a pasta Arduino - Programas

Chama o teu professor para avaliar.