Capítulo 1 — Conhecendo o Arduino

Objetivos deste capítulo

- Entender o que é um microcontrolador e o que é o Arduino
- Identificar as partes da placa Arduino UNO
- Aprender a usar a protoboard
- Conhecer os componentes do kit e o código de cores dos resistores
- Instalar a Arduino IDE e montar o primeiro circuito

1.1 O que é um microcontrolador?

Um **microcontrolador** é um computador completo dentro de um único chip: ele tem processador, memória e "portas" para se comunicar com o mundo exterior. A diferença para o computador que você usa é o tamanho e a potência: o microcontrolador é muito mais simples e barato, mas gasta pouquíssima energia e é perfeito para uma tarefa específica — controlar um aparelho.

	Computador comum	Microcontrolador do Arduino UNO
Processador	vários GHz	16 MHz
Memória RAM	vários GB	2 KB (sim, quilobytes!)
Sistema operacional	Windows, Linux, macOS	nenhum — roda só o seu programa
Preço do chip	centenas de reais	alguns reais
Consumo	dezenas de watts	menos de 1 watt

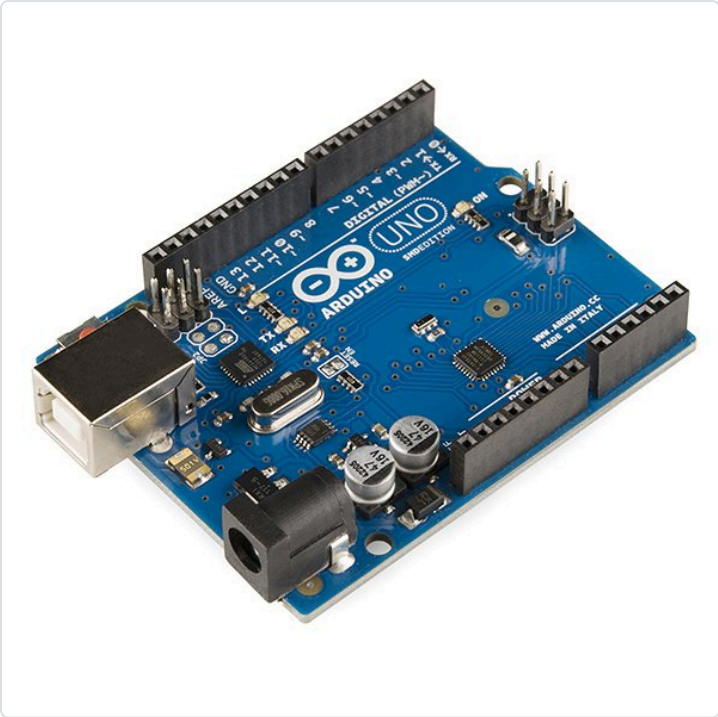
Parece fraco? Para piscar LEDs, ler sensores e mover motores, é mais do que suficiente. É exatamente esse tipo de chip que está dentro do seu micro-ondas, do controle remoto e do painel do carro.

1.2 O que é o Arduino?

O **Arduino** é uma placa que torna fácil usar um microcontrolador. Ela foi criada na Itália em 2005 por professores que queriam que estudantes de design — sem formação em engenharia — conseguissem construir protótipos interativos.

A placa traz o microcontrolador já montado com tudo o que ele precisa para funcionar: regulador de energia, conexão USB para o computador, e pinos onde você encaixa fios sem precisar soldar. Junto com a placa vem a **Arduino IDE**, o programa onde você escreve o código e o envia para a placa.

O Arduino é **open source**: seus projetos são abertos, e por isso existem centenas de placas compatíveis e uma comunidade enorme de pessoas compartilhando projetos e tirando dúvidas.

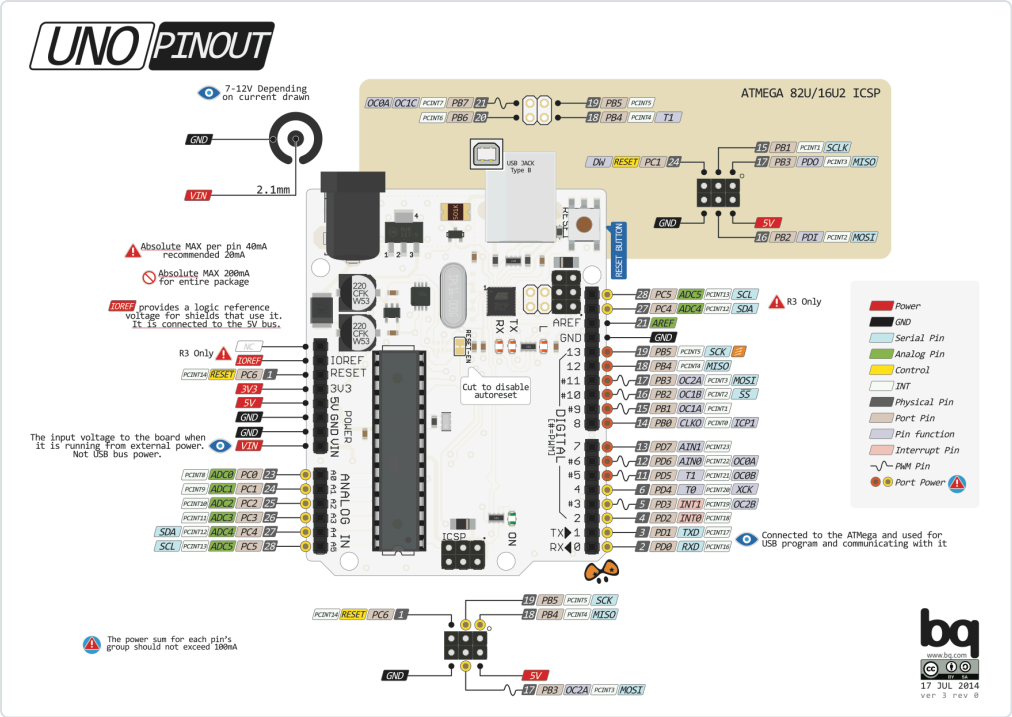


A placa Arduino UNO R3. O conector USB fica à esquerda; os pinos digitais, na parte de cima; os analógicos e de alimentação, embaixo.

1.3 A placa Arduino UNO

Vamos conhecer a placa que você vai usar. Pegue o Arduino na mão e localize cada parte:

```
                ┌──────────────────────────────────────────────┐
Conector        │ [USB]                  ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ │  Pinos DIGITAIS
USB  ──────►    │                       13 12 ~11 ~10 ~9 8 7 ~6 ~5 4 ~3 2 1 0 │  (0 a 1
                │        [LED L]  [LED ON]                     │  ~ = também faz PWM
                │  [RESET]                                     │
Conector        │ [JACK]     [ATmega328P]                      │
de energia      │           ○ ○ ○ ○ ○ ○ ○ ○     ○ ○ ○ ○ ○ ○     │
(7-12 V)        │        3.3V 5V GND GND Vin   A0 A1 A2 A3 A4 A5 │  Pinos ANALÓGICOS
                └──────────────────────────────────────────────┘
                          Alimentação            (A0 a A5)
```



Mapa completo dos pinos do Arduino UNO. Não se preocupe com tantos nomes: nesta apostila você vai usar apenas os pinos digitais (0-13), os analógicos (A0-A5) e os de alimentação (5V, 3.3V, GND).

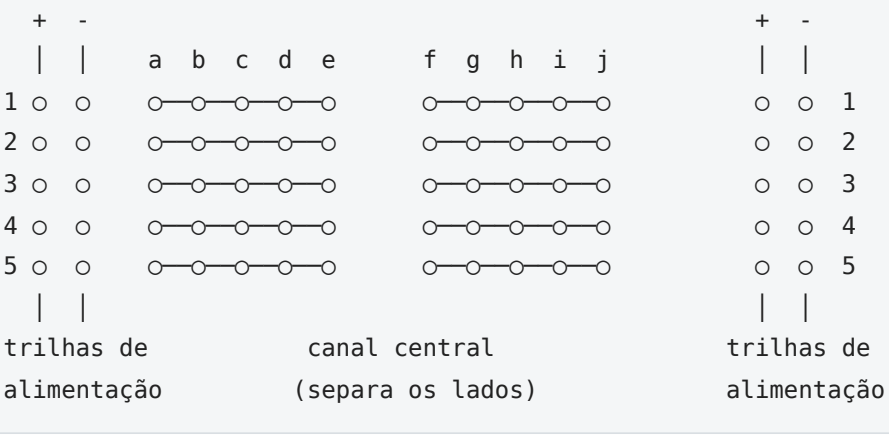
As partes mais importantes:

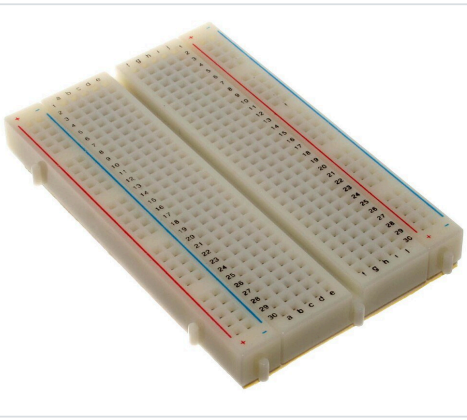
- **Conector USB** — liga a placa ao computador. Serve para enviar o programa *e* para alimentar a placa. Na maioria dos projetos desta apostila, o USB é tudo o que você precisa.
- **Conector de energia (jack)** — para usar a placa sem o computador, com uma fonte de 7 a 12 V ou uma bateria.
- **Pinos digitais (0 a 13)** — funcionam como interruptores que você controla pelo código: podem estar **ligados (5 V)** ou **desligados (0 V)**. Também servem para *ler* se algo está ligado ou desligado (um botão, por exemplo). Os pinos marcados com **~** (3, 5, 6, 9, 10, 11) fazem algo a mais chamado PWM, que veremos no capítulo 7.
- **Pinos analógicos (A0 a A5)** — leem valores que variam continuamente, como a posição de um potenciômetro ou a quantidade de luz em um sensor.
- **Pinos de alimentação** — **5V** e **3.3V** fornecem energia para os componentes; **GND** (*ground*, terra) é o "zero volts", o ponto de referência de todo circuito. Há vários pinos GND e todos são iguais.
- **Botão RESET** — reinicia o programa do zero.
- **LED L** — um pequeno LED ligado ao pino 13. Ele será seu melhor amigo no capítulo 2.
- **LED ON** — acende sempre que a placa está alimentada.
- **Microcontrolador ATmega328P** — o chip preto e comprido. É onde seu programa roda.

> ⚠️ Os pinos **0 (RX) e 1 (TX)** são usados pela comunicação USB. Evite ligar componentes neles — seus projetos podem parar de receber programas.

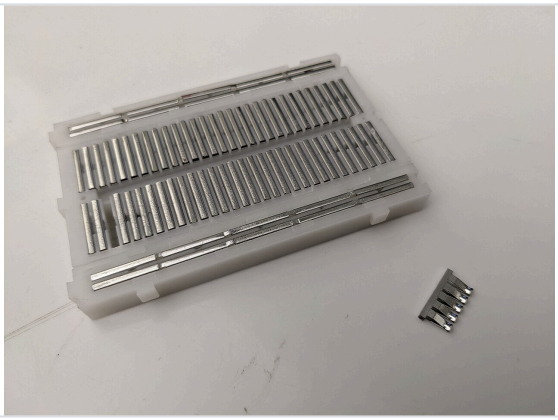
1.4 A protoboard

A **protoboard** (ou *matriz de contatos*) é uma placa cheia de furos onde você encaixa componentes e fios sem soldar. O segredo é saber **quais furos estão conectados entre si por baixo**.





Uma protoboard de 400 pontos, vista de cima.



O que há por baixo: tiras de metal que conectam os furos — grupos de 5 na área central e linhas compridas nas trilhas laterais.

Regras da protoboard:

1. **Nas linhas numeradas (a área central)**, cada grupo de **5 furos na horizontal** (a-b-c-d-e, ou f-g-h-i-j) está conectado entre si. O canal do meio separa os dois lados: `e` **não** está ligado a `f`.
2. **Nas trilhas laterais** (marcadas com + e −, geralmente com linhas vermelha e azul), todos os furos de uma **coluna vertical** estão conectados. Usamos a trilha + para o 5 V e a trilha − para o GND.
3. Linhas diferentes (1, 2, 3...) **não** estão conectadas entre si.

⚠️ Em algumas protoboards maiores, a trilha lateral é dividida no meio. Se algo não funciona, verifique se o fio e o componente estão no mesmo trecho da trilha.

1.5 Os componentes do kit

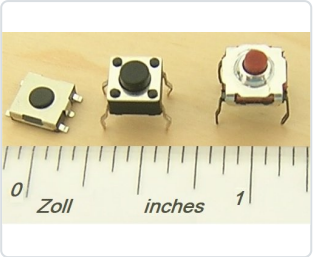
Componente	Para que serve	Como identificar
LED	Emite luz. É uma *saída*: o Arduino o liga e desliga.	Tem uma perna mais comprida (**+**, ânodo) e uma mais curta (**−**, cátodo).
Resistor	Limita a corrente elétrica, protegendo LEDs e outros componentes.	Cilindro pequeno com faixas coloridas. Não tem lado certo.
Botão (push button)	Fecha o circuito enquanto está pressionado. É uma *entrada*.	Quadradinho com 4 pernas.
Potenciômetro	Resistor variável — um botão giratório. Entrada analógica.	Tem 3 pernas e um eixo para girar.
LDR	Resistor que muda com a luz. Sensor de luminosidade.	Disco com uma linha ondulada em cima.
Buzzer	Emite som quando recebe sinal. Saída.	Cilindro preto com um furo em cima. Tem + e −.
Servo motor	Motor que gira para um ângulo exato (0° a 180°).	Caixinha com 3 fios (marrom, vermelho, laranja).
Sensor ultrassônico HC-SR04	Mede distância usando som, como um morcego.	Placa azul com dois "olhos".
Jumpers	Fios para ligar tudo.	Fios coloridos com pontas de metal.



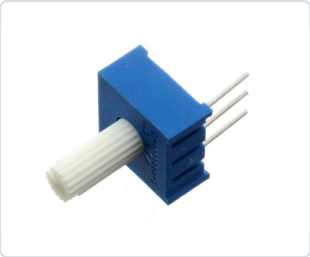
LED de 5 mm. A perna mais comprida é o +.



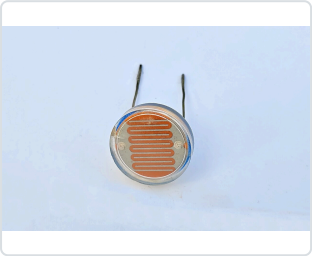
Resistores: as faixas coloridas indicam o valor.



Botões (push buttons) de 4 pernas.



Potenciômetro de 10 kΩ para protoboard.



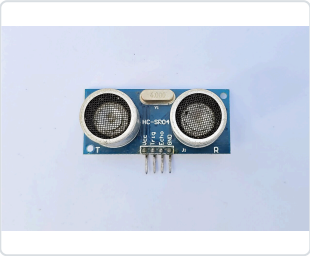
LDR: sensor de luz.



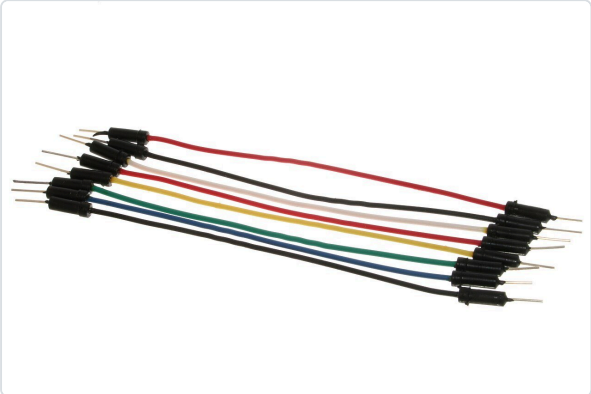
Buzzer piezoelétrico.



Servo motor SG90.



Sensor ultrassônico HC-SR04.



Jumpers: fios com pontas rígidas, feitos para encaixar na protoboard e nos pinos do Arduino.

Entradas e saídas

Repare que os componentes se dividem em dois grupos. Essa é a ideia central de todo projeto com Arduino:

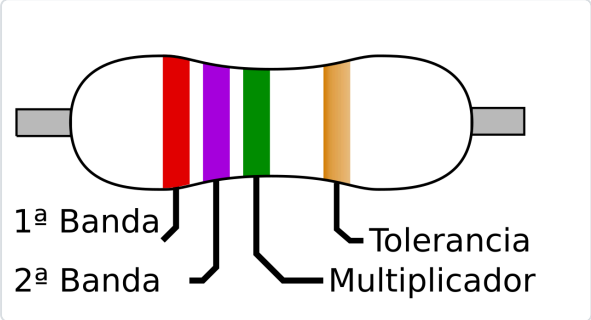
- **Entradas** (*inputs*) — levam informação do mundo para o Arduino: botões, potenciômetro, LDR, sensor ultrassônico.
- **Saídas** (*outputs*) — levam ações do Arduino para o mundo: LEDs, buzzer, servo.

Um programa de Arduino, no fundo, é sempre isto: **ler as entradas, decidir algo, acionar as saídas.**

1.6 Lendo o valor de um resistor

Os resistores têm faixas coloridas que indicam seu valor em **ohms (Ω)**. Para resistores de 4 faixas, as duas primeiras são dígitos, a terceira é o multiplicador (quantos zeros acrescentar) e a quarta é a tolerância (geralmente dourada).

Cor	Dígito	Multiplicador
Preto	0	×1
Marrom	1	×10
Vermelho	2	×100
Laranja	3	×1.000
Amarelo	4	×10.000
Verde	5	×100.000
Azul	6	×1.000.000
Violeta	7	—
Cinza	8	—
Branco	9	—



Como ler as faixas: as duas primeiras são dígitos, a terceira é o multiplicador e a última (dourada ou prateada) é a tolerância. Segure o resistor com a faixa dourada à direita.

Exemplos que você vai usar o tempo todo:

- **Vermelho – Vermelho – Marrom** → 2, 2, ×10 = **220 Ω** (para LEDs)
- **Marrom – Preto – Laranja** → 1, 0, ×1.000 = **10.000 Ω = 10 kΩ** (para botões e sensores)

> 💡 Na dúvida, há aplicativos de celular e sites que calculam o valor pela cor. Com o tempo você vai reconhecer os mais comuns de olho.

1.7 Regras de segurança

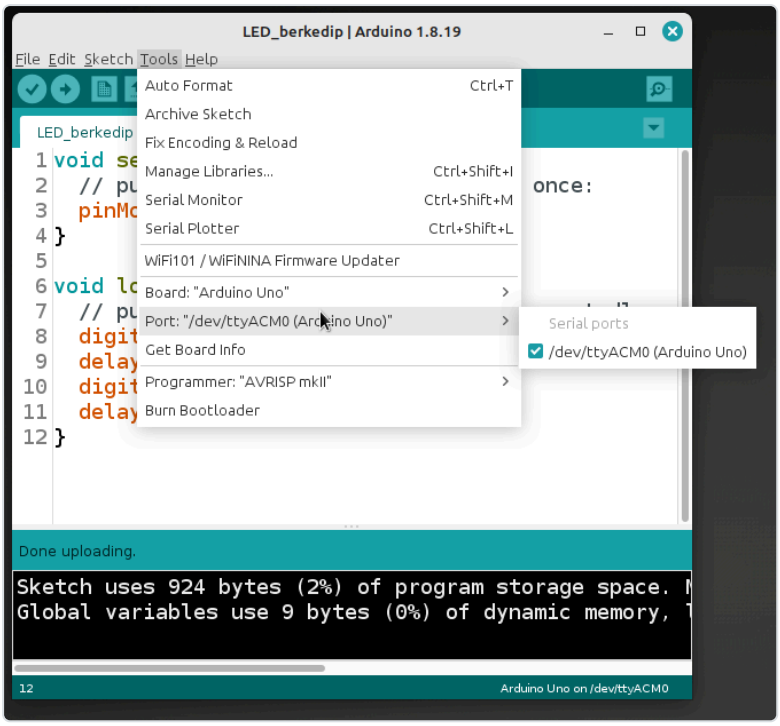
O Arduino é bastante resistente, mas alguns erros podem queimar a placa ou um componente. Grave estas regras:

1. **Nunca ligue o 5V diretamente no GND** com um fio. Isso é um curto-circuito.
2. **Nunca ligue um LED sem resistor.** Ele pode queimar na hora (ou danificar o pino).
3. **Desconecte o USB antes de mexer no circuito.** Monte tudo, confira, e só então conecte.
4. **Nunca ligue um pino digital diretamente em outro pino digital** nem no 5V/GND sem resistor.
5. Não coloque a placa sobre superfícies metálicas.
6. Motores e servos puxam muita corrente. Um servo pequeno funciona no USB, mas mais que um pode exigir fonte externa.

1.8 Instalando a Arduino IDE

A **IDE** (*Integrated Development Environment*) é o programa onde você escreve o código e o envia à placa.

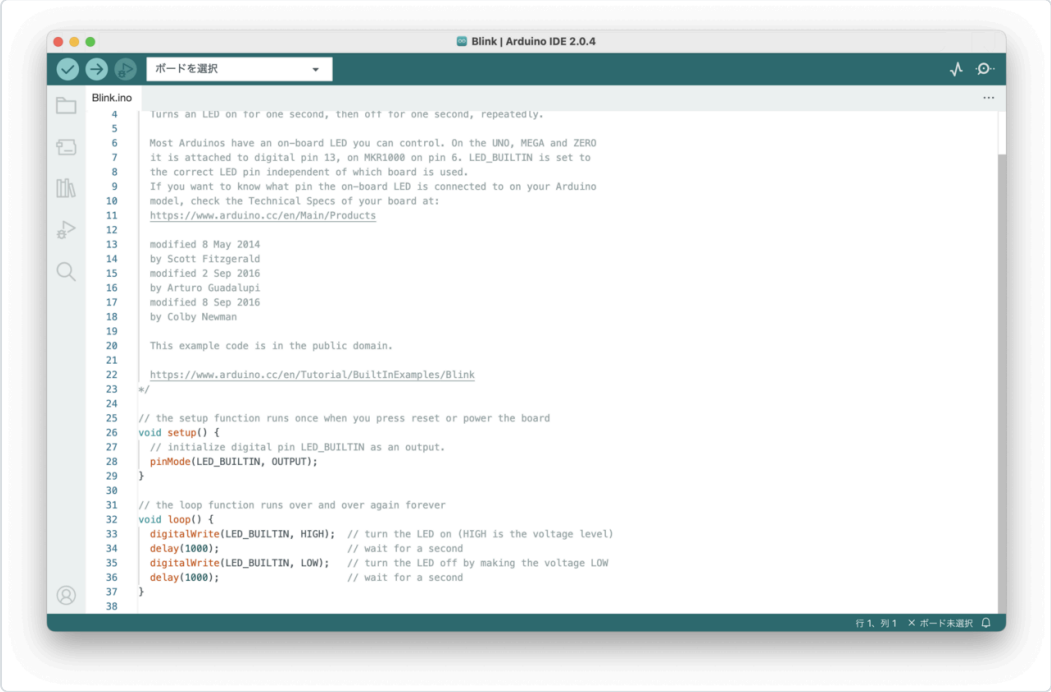
1. Acesse arduino.cc/en/software e baixe a versão para o seu sistema (Windows, macOS ou Linux). Instale normalmente.
2. Conecte o Arduino ao computador com o cabo USB. O LED ON deve acender.
3. Abra a IDE. No menu superior, escolha a placa: **Tools → Board → Arduino UNO** (ou *Ferramentas → Placa*).
4. Escolha a porta: **Tools → Port** — aparece algo como `COM3` (Windows) ou `/dev/ttyUSB0` (Linux) ou `/dev/cu.usbmodem…` (macOS). Se não souber qual é, desconecte o Arduino, veja quais portas aparecem, conecte de novo e veja qual surgiu.



Selecionando a porta no menu Tools da IDE. A porta certa é a que mostra "(Arduino Uno)" ao lado.

> 💡 **Placas compatíveis** (com chip CH340) às vezes precisam de um driver adicional no Windows. Se a porta não aparece, pesquise "driver CH340".

As partes da IDE



A Arduino IDE 2 com o programa Blink aberto. Os botões de Verificar (✓) e Enviar (→) ficam no canto superior esquerdo.

```
┌──────────────────────────────────────────────┐
│ [✓] [→]   [▼ Arduino UNO em COM3]      [🔍]  │  ← Verificar, Enviar, Placa, Monitor S
├──────────────────────────────────────────────┤
│ void setup() {                               │
│                                              │  ← Editor: aqui você escreve o código
│ }                                            │
│                                              │
│ void loop() {                                │
│                                              │
│ }                                            │
├──────────────────────────────────────────────┤
│ Saída / Console                              │  ← Mensagens de erro e de sucesso
└──────────────────────────────────────────────┘
```

- ✓ **Verificar (Verify)** — confere se o código está correto, sem enviar.
- → **Enviar (Upload)** — compila e envia o programa para a placa.
- 🔍 **Monitor Serial** — janela para o Arduino "conversar" com o computador (capítulo 6).

1.9 🔧 Mão na massa: seu primeiro circuito (sem código!)

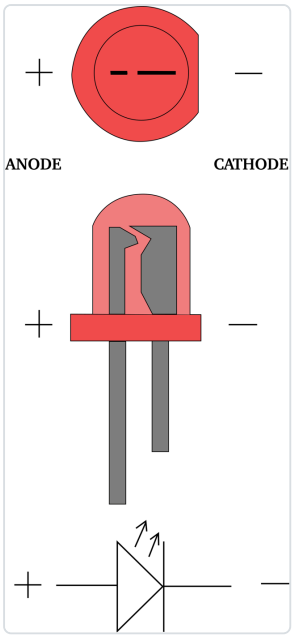
Vamos acender um LED usando apenas a energia da placa, para você se acostumar com a protoboard.

Material: 1 LED, 1 resistor de 220 Ω, 2 jumpers.

Montagem:

De	Para
Pino **5V** do Arduino	Trilha **+** da protoboard
Pino **GND** do Arduino	Trilha **−** da protoboard
Perna **comprida (+)** do LED	Linha 5, furo `a`
Perna **curta (−)** do LED	Linha 6, furo `a`
Resistor	Linha 5, furo `c` até a trilha **+**
Jumper	Linha 6, furo `c` até a trilha **−**

```
5V ─── trilha + ─── resistor 220Ω ─── LED(+) ── LED(−) ─── trilha − ─── GND
```



O LED tem polaridade: a perna comprida (ânodo, +) recebe a corrente; a curta (cátodo, −) vai para o GND. O lado achatado da borda também marca o −.

Confira tudo, conecte o USB e... o LED deve acender! Se não acendeu:

- O LED está invertido? Tente virar.
- O resistor e o LED estão na **mesma linha** da protoboard?
- Os jumpers de 5V e GND estão nas trilhas certas?

> 💡 A corrente sai do 5V, passa pelo resistor, entra no LED pela perna comprida, sai pela perna curta e volta ao GND. Se qualquer trecho desse caminho estiver interrompido, nada acontece. Isso se chama **circuito fechado**.

📝 Para fixar

1. Qual a diferença entre um pino digital e um pino analógico?
2. Para que serve o pino GND?
3. Na protoboard, os furos `a1`, `b1` e `c1` estão conectados entre si? E `a1` e `a2`? E `e1` e `f1`?
4. Qual o valor de um resistor com as faixas **Laranja – Laranja – Marrom**?
5. Classifique como **entrada** ou **saída**: buzzer, LDR, servo motor, botão, LED, sensor ultrassônico.
6. Por que nunca se deve ligar um LED sem resistor?