



## >> capítulo 1

# Por onde começo?

*Este capítulo é para o leitor que está ansioso para começar. A sua placa nova de Arduino chegou e você quer que ela faça alguma coisa. Portanto, mãos à obra.*

### **Objetivos deste capítulo**

- >> Explicar como energizar o Arduino.
- >> Demonstrar a instalação do ambiente de desenvolvimento do Arduino no Windows 7, no Mac OS X e no Linux.
- >> Aplicar a configuração do ambiente do Arduino.
- >> Demonstrar como modificar o Sketch Blink.
- >> Ensinar a utilizar um resistor e um LED externo de maior potência.
- >> Auxiliar no reconhecimento um protoboard.

## » Alimentação elétrica

Quando você compra uma placa de Arduino Uno ou Leonardo, ela costuma vir com um programa Blink (Piscar) pré-instalado que fará piscar o pequeno LED existente na placa. A Figura 1-1 mostra duas placas Arduino.

O diodo emissor de luz (LED), assinalado com a letra L na placa, está ligado a um dos pinos de conexão de entrada e saída digitais da placa. Ele está conectado ao pino digital 13. Isso limita o uso do pino 13 a ser uma saída. No entanto, o LED consome apenas uma pequena quantidade de corrente, de modo que você ainda pode conectar outras coisas a esse pino de conexão.

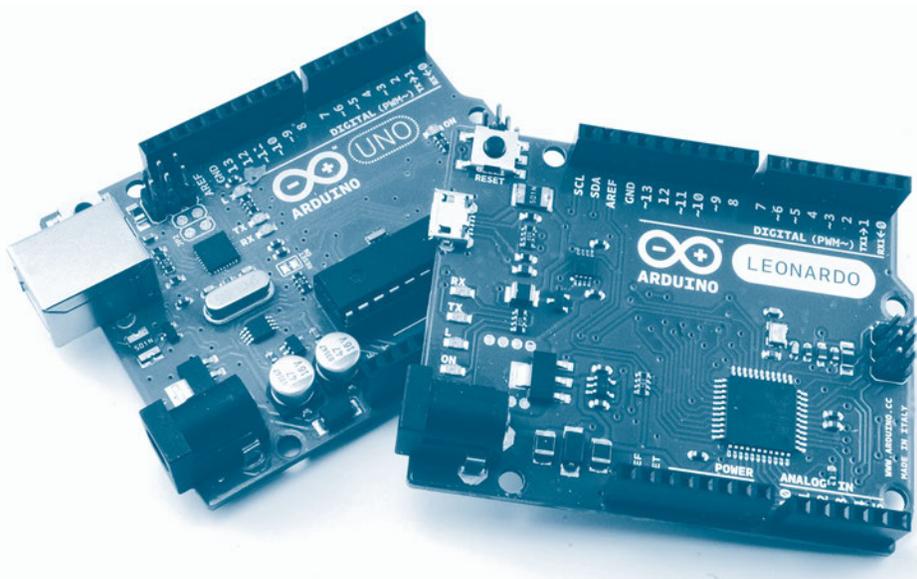
Tudo que você precisa para colocar o seu Arduino em funcionamento é alimentá-lo com energia elétrica. A maneira mais fácil de fazer isso é conectá-lo a uma porta USB (*Universal Serial Bus*) do seu computador. Para um Arduino Uno, você precisará de um cabo USB do tipo A-B. Esse é o mesmo tipo de cabo que normalmente é usado para conectar um computador a uma impressora. Para um Leonardo, você precisará de um conector USB do tipo micro. É

possível que surjam algumas mensagens do seu sistema operacional a respeito de novos dispositivos ou hardware encontrados. Por enquanto, ignore-os.

Se tudo estiver funcionando corretamente, o LED deverá piscar (blink) uma vez a cada dois segundos. Nas placas novas de Arduino, esse sketch (Blink), que faz o LED piscar, já vem instalado e é utilizado para verificar se a placa está funcionando. Se você clicar no botão de Reset, o LED deverá piscar momentaneamente. Se esse for o caso e o LED não piscar, então provavelmente a placa não veio programada com o sketch Blink (piscar). Não se desespere pois, depois de tudo instalado, faremos modificações e instalaremos esse sketch como nosso primeiro projeto.

## » Como instalar o software

Agora que o Arduino está funcionando, vamos instalar o software de modo que possamos modificar o programa Blink (piscar) e enviá-lo à placa. O procedimento exato depende do sistema operacional que você está utilizando no seu computador. No entanto, o princípio básico é o mesmo para todos.



**Figura 1-1** Arduinos Uno e Leonardo.  
Fonte: do autor.

Instale o ambiente de desenvolvimento do Arduino. Esse é o programa que você executa no seu computador permitindo que você escreva sketches e transfira-os para a placa do Arduino.

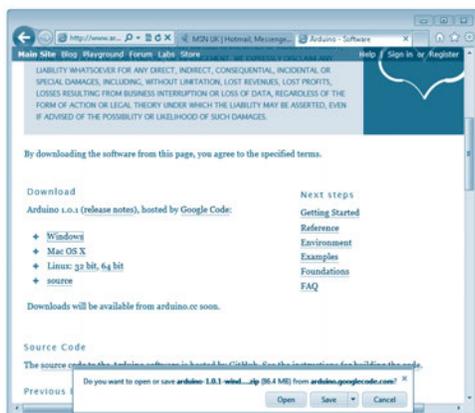
Instale o driver USB que permite a comunicação do computador com a porta USB do Arduino. Isso é usado para a programação e o envio de mensagens.

O site do Arduino ([www.arduino.cc](http://www.arduino.cc)) contém a última versão do software. Neste livro, usaremos a versão Arduino 1.0.2.

## » Instalação no Windows

As instruções seguintes são para a instalação no Windows 7. O processo é quase o mesmo para o Windows Vista e XP. A única parte que pode ser um pouco trabalhosa é a instalação dos drivers.

Siga as instruções de como baixar o arquivo para Windows na página inicial do Arduino ([www.arduino.cc](http://www.arduino.cc)). Com isso, você baixará o arquivo com extensão Zip contendo o software de Arduino, como está mostrado na Figura 1-2. É possível que a versão baixada do software seja mais recente do que a versão 1.0.2 mostrada (a equipe de desenvolvimento do Arduino ainda não atualizou o nome do arquivo Zip).



**Figura 1-2** Baixando o software de Arduino para Windows.

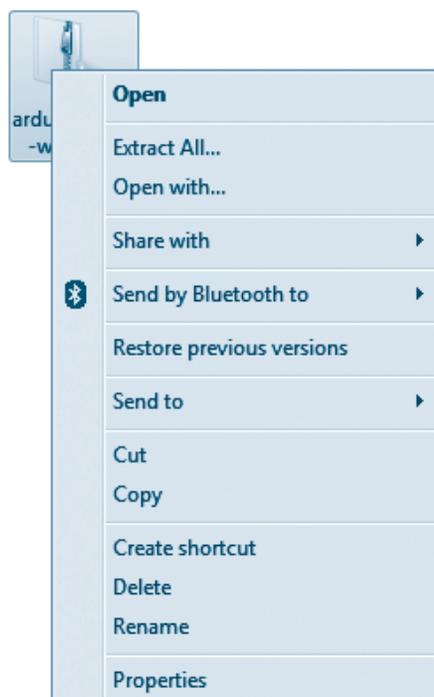
Fonte: do autor.

O software de Arduino não faz distinção entre as diversas versões de Windows. O arquivo baixado deve funcionar com todas as versões, de Windows XP em diante. As instruções seguintes são para o Windows 7.

Selecione a opção “Salvar” na caixa de diálogo e salve o arquivo Zip no seu Desktop. A pasta contida no arquivo Zip se tornará a pasta principal de Arduino. Agora descompacte o arquivo (unzip) no seu Desktop. Mais tarde, se desejar, você poderá movê-lo para algum outro local.

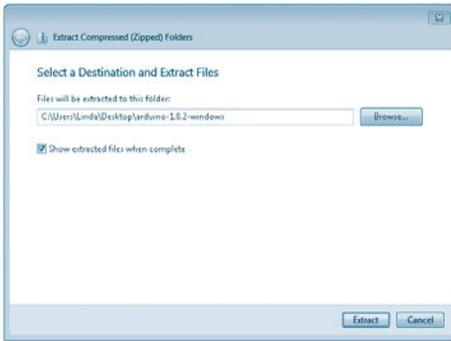
No Windows, você pode fazer isso clicando no arquivo Zip com o botão direito. Então aparecerá o menu da Figura 1-3. Em seguida selecione a opção “Extrair Tudo” (Extract All) para abrir o assistente de extrair pastas, mostrado na Figura 1-4.

Extraia os arquivos transferindo-os para o seu Desktop.



**Figura 1-3** A opção “Extrair Tudo” (Extract All) do menu no Windows.

Fonte: do autor.



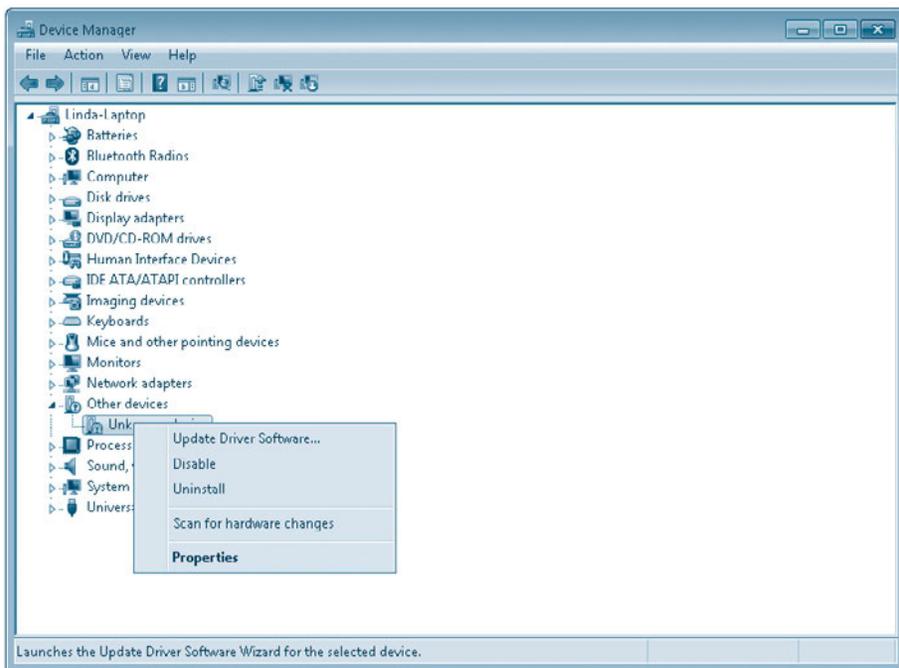
**Figura 1-4** Extração do arquivo Arduino no Windows.  
*Fonte: do autor.*

Isso criará uma nova pasta para essa versão de Arduino (no caso, 1.0.2) no seu Desktop. Se desejar, você poderá ter diversas versões de Arduino instaladas ao mesmo tempo, cada uma na sua própria pasta. Atualizações do Arduino não são muito frequentes e historicamente têm sempre mantido uma boa compatibilidade com versões anteriores do software. Portanto, a menos que você esteja tendo problemas ou que haja um novo recurso de

software que você gostaria de utilizar, não é essencial manter-se atualizado com a última versão.

Agora que você instalou a pasta de Arduino no lugar correto, precisamos instalar os drivers USB. Se você ainda não o fez, conecte o seu Leonardo ou Uno no seu computador. Dependendo da versão do seu Windows, poderão ocorrer algumas tentativas não muito bem-sucedidas por parte do seu sistema operacional para instalar os drivers. Simplesmente cancele essas tentativas na primeira vez que surgirem porque provavelmente não funcionarão. Em vez disso, abra o Gerenciador de Dispositivos. Ele pode ser acessado de diversas maneiras, dependendo da sua versão de Windows. No Windows 7, primeiro você precisa abrir o Painel de Controle. A seguir, selecione a opção para ver ícones e o Gerenciador de Dispositivos deverá aparecer na lista.

Na seção “Outros Dispositivos,” você deverá ver um ícone em “Dispositivo Desconhecido” com um pequeno triângulo amarelo de alerta. Esse ícone é do seu Arduino (Figura 1-5).



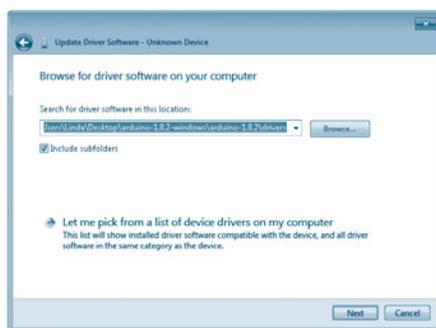
**Figura 1-5** O Gerenciador de Dispositivos do Windows.  
*Fonte: do autor.*

Clique com o botão direito em “Dispositivo Desconhecido” e selecione a opção “Atualizar Driver”. Então, você deverá escolher entre “Pesquisar automaticamente software de driver atualizado” ou “Procurar software de driver no computador”. Selecione a segunda opção e navegue até “arduino-1.0.2-windows\arduino1.0.2\drivers” (Figura 1-6). Modifique os números da versão se você estiver usando uma versão diferente de Arduino.

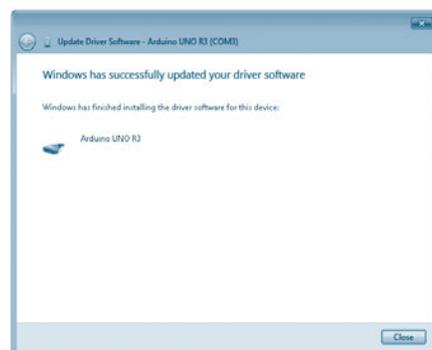
Clique em “Avançar”. Poderá surgir um alerta de segurança. Se isso ocorrer, permita a instalação do software. Depois da instalação do software, aparecerá uma mensagem de confirmação como a mostrada na Figura 1-7. Para um Leonardo, a mensagem será diferente, mas o procedimento é idêntico.

Agora, o Gerenciador de Dispositivos deve mostrar o nome correto para o Arduino (Figura 1-8).

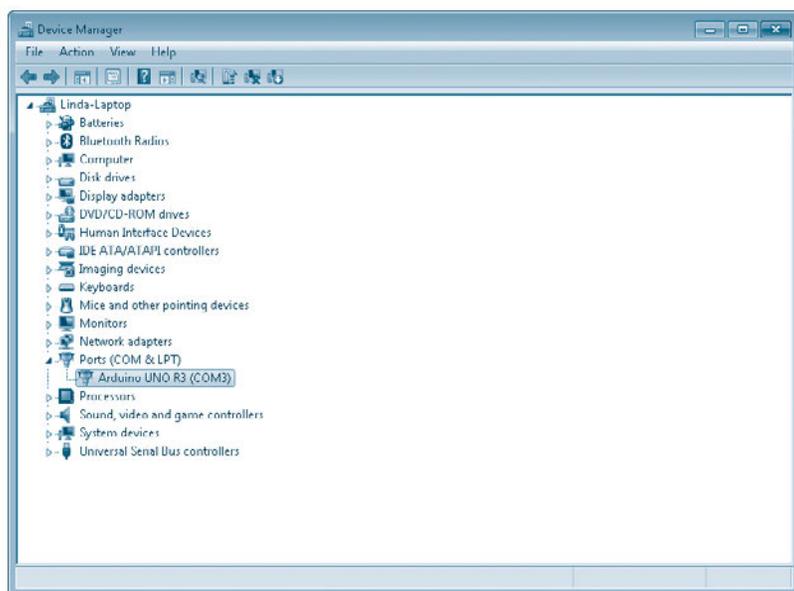
Esse processo é realizado uma única vez. De agora em diante, sempre que você conectar a sua placa de Arduino, os seus drivers USB serão carregados automaticamente e o Arduino estará pronto para funcionar.



**Figura 1-6** Buscando os drivers USB.  
Fonte: do autor.



**Figura 1-7** A instalação bem-sucedida do driver USB.  
Fonte: do autor.



**Figura 1-8** O Gerenciador de Dispositivos mostrando o Arduino.  
Fonte: do autor.

## »» Instalação no Mac OS X

O processo de instalação do software de Arduino no Mac é bem mais fácil do que no PC.

Como antes, o primeiro passo é baixar o arquivo. No caso do Mac, é um arquivo Zip. Depois disso, clique duas vezes no arquivo Zip, do qual será extraído um arquivo simples denominado "Arduino.app". Esse arquivo contém o aplicativo Arduino completo. Basta arrastá-lo à sua pasta de aplicativos.

Agora, você pode localizar e executar o software do Arduino que está na pasta de aplicativos. Como você irá usá-lo frequentemente, talvez queira clicar com o botão direito no ícone do Arduino que está no "Dock" e ativar "Keep in Dock" (mantenha no dock).

## »» Instalação no Linux

Há muitas distribuições diferentes de Linux, e as instruções para cada distribuição são ligeiramente diferentes. A comunidade de Arduino tem feito um grande trabalho ao reunir conjuntos de instruções para cada distribuição. Dessa forma, sugerimos que você acesse, por exemplo, o site <http://playground.arduino.cc/learning/linux> e siga as orientações para a distribuição que você usa.

## »» Configurando o ambiente do seu Arduino

Independentemente do tipo de computador que você está usando, agora o software de Arduino já deve estar instalado, sendo necessário fazer alguns ajustes. Você precisa especificar a porta serial que está conectada à placa de Arduino e também o tipo de placa de Arduino que você está utilizando. Mas, primeiro, você precisa conectar o seu Arduino ao computador por meio do cabo USB. Se não fizer isso, não será possível selecionar a porta serial.

A seguir, execute o software de Arduino. No Windows, isso significa abrir a pasta "Arduino" e clicar no ícone "Arduino" (selecionado na Figura 1-9). Se você preferir, pode colocar um atalho na sua Área de Trabalho.

A porta serial é configurada no menu Ferramentas, como mostrado na Figura 1-10 no Mac e na Figura 1-11 no Windows 7 – a lista de portas no Linux é similar à do Mac.

Se você estiver utilizando muitos dispositivos USBs ou Bluetooth no seu Mac, é provável que, na lista de portas, haja poucas opções disponíveis. Selecione o item da lista que começa por "dev/tty.usbserial".

No Windows, a porta serial pode ser configurada comumente para COM3 ou COM4, dependendo do que aparecer.

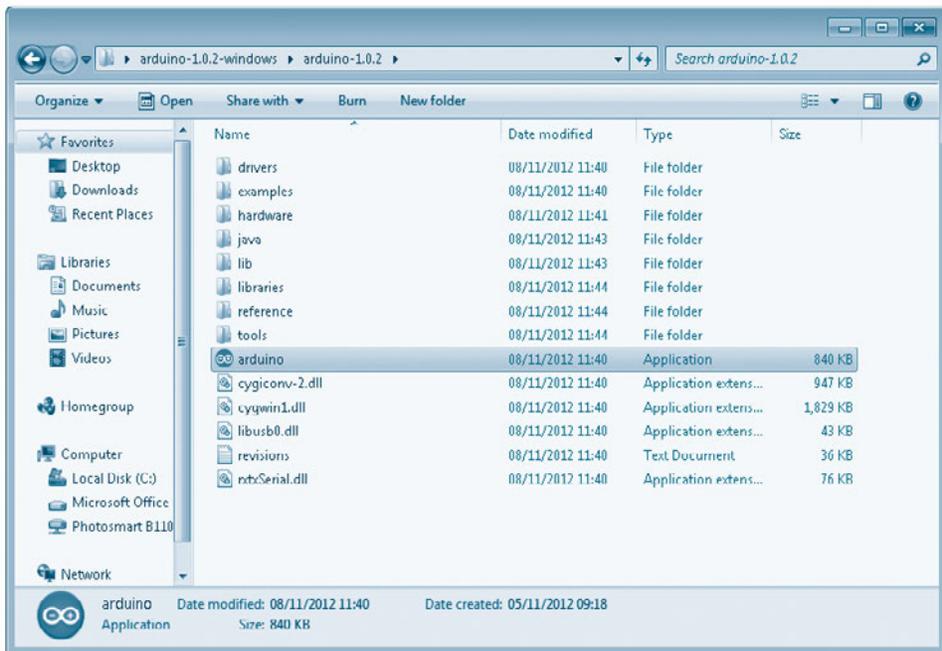
No menu Ferramentas, agora podemos escolher a placa que será utilizada, como mostrado na Figura 1-12.

## »» Como baixar o software para os projetos

Os sketches (como são denominados os programas no mundo do Arduino) usados neste livro estão disponíveis para download na forma de um arquivo simples Zip com menos de um megabyte. Portanto, você pode baixar o software para todos os projetos, mesmo que você pretenda usar apenas alguns deles. Para baixá-los, acesse o site [www.simonmonk.org/](http://www.simonmonk.org/) e siga os links de download para a segunda edição\* deste livro.

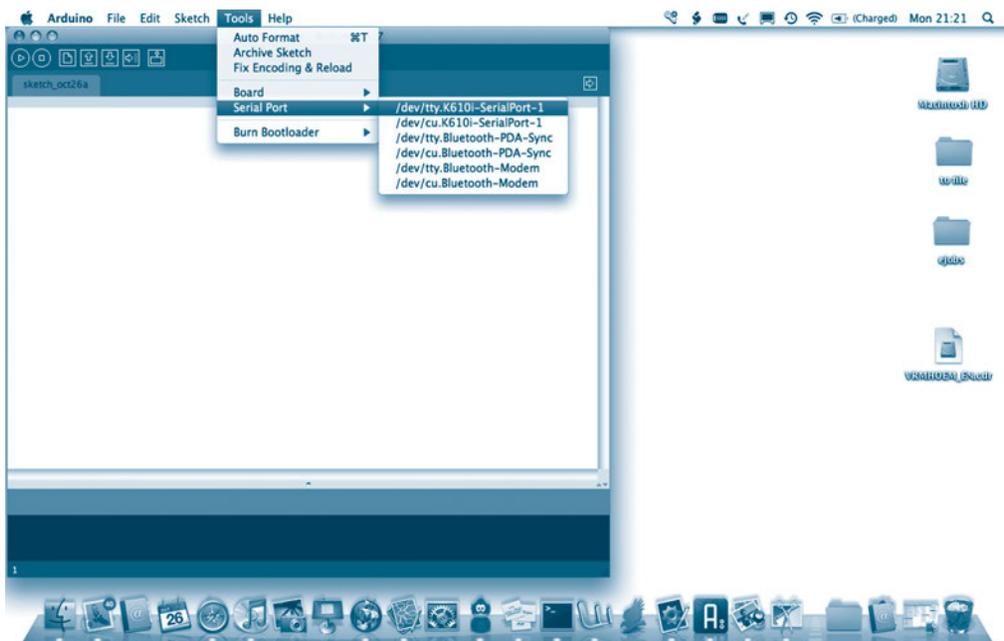
Com qualquer plataforma, o software do Arduino espera encontrar todos os sketches na pasta "Meus Documentos", dentro de um diretório denominado "Arduino", que foi criado pelo software do Arduino

\* N. de T.: É o livro no qual a expressão "Second edition", quase ilegível, está acima de "30 Arduino".



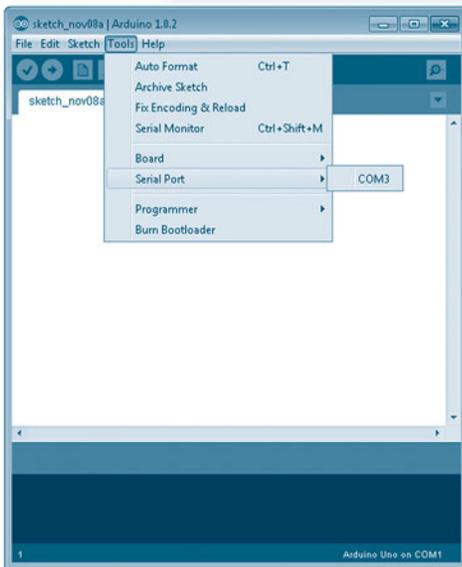
**Figura 1-9** Iniciando a execução do Arduino no Windows.

Fonte: do autor.

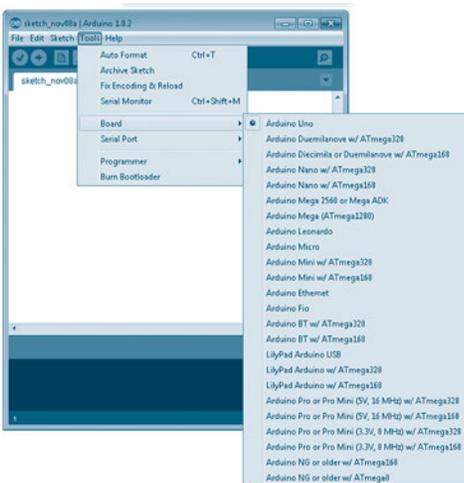


**Figura 1-10** Configurando a porta serial no Mac.

Fonte: do autor.



**Figura 1-11** Configurando a porta serial no Windows.  
Fonte: do autor.



**Figura 1-12** Configurando a placa.  
Fonte: do autor.

na primeira vez em que foi executado. Portanto, coloque os conteúdos do arquivo Zip dentro dessa pasta, supondo que você já tenha instalado o software do Arduino.

Observe que os sketches são numerados por projeto e vêm em suas próprias pastas.

## Projeto 1

### >> LED piscante

Supondo que você já tenha instalado com sucesso o software, agora podemos iniciar o nosso primeiro projeto emocionante. Na verdade, ele não é tão emocionante, mas precisamos iniciar com alguma coisa e, dessa forma, poderemos verificar se configuramos tudo corretamente e se estamos prontos para utilizar o nosso Arduino.

Modificaríamos o sketch Blink (pisca) de exemplo que acompanha o Arduino. Para isso, aumentaremos a frequência do pisca-pisca e, em seguida, instalaremos o sketch modificado na placa do Arduino. Em vez de piscar lentamente, a nossa placa fará o LED piscar (flash) rapidamente. Em seguida, levaremos o projeto um passo mais adiante: utilizaremos um resistor e um LED externo de maior potência no lugar do pequeno LED que vem junto na placa do Arduino.

#### COMPONENTES E EQUIPAMENTO

	Descrição	Apêndice
	Arduino Uno ou Leonardo	m1/m2
D1	LED vermelho de 5 mm	s1
R1	Resistor de 270 Ω e 1/4 W	r3
	Protoboard	h1
	Fios de conexão (jumpers)	h2

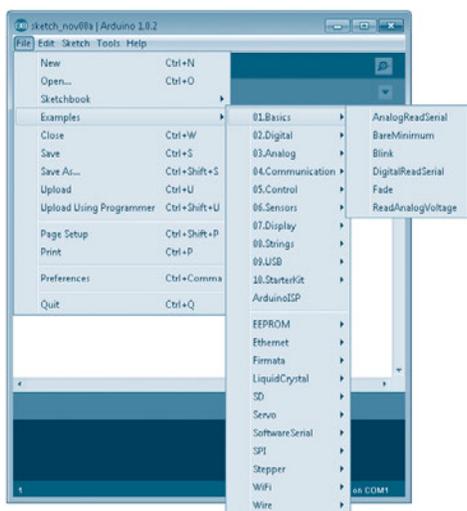
- Na realidade, praticamente qualquer LED e resistor de 270 ohms comuns serão adequados.
- O número na coluna “Apêndice” refere-se à lista de componentes disponível no Apêndice, na qual estão os respectivos códigos adotados por diversos fornecedores.

### >> Software

Primeiro, precisamos carregar o sketch Blink no software do Arduino. Esse sketch vem como exemplo quando você instala o ambiente Arduino. Portanto, podemos carregá-lo utilizando o menu (arquivo), como mostrado na Figura 1-13.

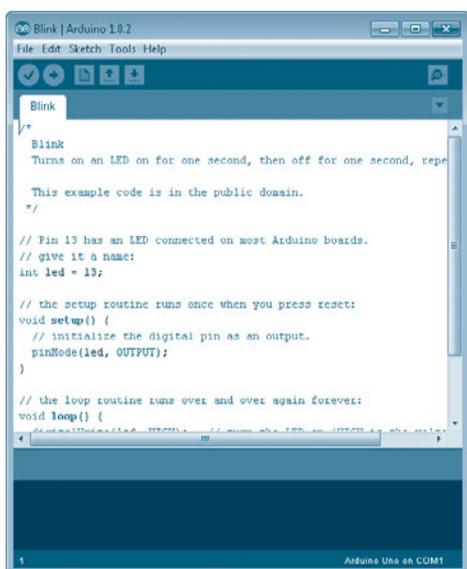
O sketch será aberto em uma janela separada (Figura 1-14). Agora, se desejar, você poderá fechar a janela vazia que foi aberta quando o software do Arduino começou a ser executado.

A maior parte do texto desse sketch está na forma de comentários. Na realidade, esses comentários



**Figura 1-13** Carregando o sketch Blink.

Fonte: do autor.



**Figura 1-14** O sketch Blink.

Fonte: do autor.

não fazem parte do programa, mas explicam o que ocorre durante a execução do programa para qualquer pessoa que estiver lendo o sketch.

Os comentários podem ter uma única linha, começando com // e estendendo-se até o final da mesma linha, ou podem ocupar diversas linhas, começando com /\* e terminando com \*/ algumas linhas depois.

Se todos os comentários de um sketch fossem retirados, o sketch ainda continuaria funcionando exatamente da mesma maneira. Nós incluímos os comentários porque são úteis a qualquer pessoa que estiver lendo o sketch tentando entender como ele funciona.

Antes de começar, é necessário uma pequena explicação. A comunidade Arduino utiliza a palavra “sketch” no lugar de “programa”. Por essa razão, de agora em diante, irei me referir aos nossos programas de Arduino como sketches. Ocasionalmente, poderei me referir a “código”. Código é um termo usado pelos programadores para fazer referência a uma seção de um programa ou, genericamente, ao que é escrito quando um programa é criado. Desse modo, alguém poderia dizer “Eu escrevi um programa que faz aquilo”, ou poderia dizer “Eu escrevi um código que faz aquilo”.

Para alterar a frequência com que o LED pisca, precisamos modificar o valor do retardo (delay). Para isso, nos dois lugares do sketch em que temos

```
delay(1000);
```

devemos trocar o valor dentro dos parênteses por 200, de modo que teremos

```
delay(200);
```

Isso alterará o retardo (delay) entre o ligar e o desligar do LED. Esse tempo passará de 1.000 milissegundos (1 segundo) para 200 milissegundos (um quinto de segundo). No Capítulo 3, exploraremos com mais profundidade esse sketch. Por enquanto, vamos simplesmente alterar o retardo e transferir (upload) o sketch para a placa de Arduino.

Com a placa conectada ao computador, clique no botão "Upload" do software do Arduino. Isso está mostrado na Figura 1-15. Se tudo estiver funcionando corretamente, haverá uma pequena pausa e, em seguida, os dois LEDs vermelhos da placa piscarão de forma intermitente enquanto o sketch estiver sendo transferido (uploading) para a placa. Isso deverá levar uns 5 a 10 segundos.

Se isso não ocorrer, verifique a porta serial e as configurações do tipo de placa de Arduino que está sendo utilizado, como foi descrito nas seções anteriores.

Quando o sketch completo estiver instalado, a placa será inicializada (reset) automaticamente e, se tudo estiver certo, você verá o LED da porta digital 13 começar a piscar mais rapidamente do que antes.

## »» Hardware

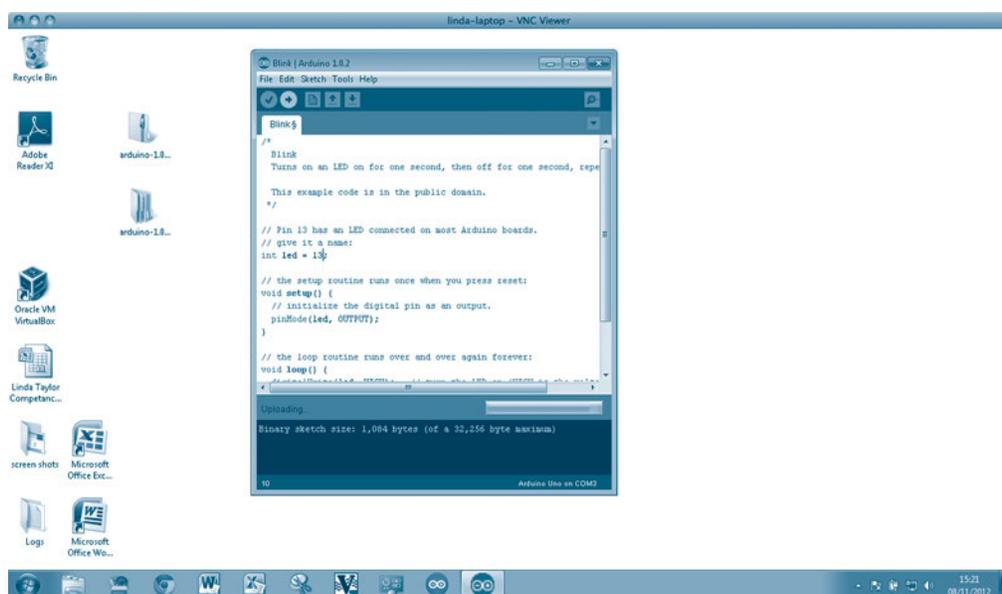
Até agora, parece que ainda não trabalhamos com eletrônica de verdade, porque o hardware estava todo contido na placa do Arduino. Nesta seção, acrescentaremos um LED externo à placa.

Não podemos simplesmente aplicar uma tensão diretamente aos LEDs. Eles devem ser conectados a um resistor limitador de corrente. O LED e o resistor podem ser encomendados de qualquer fornecedor de material eletrônico. Os códigos desses componentes para diversos fornecedores estão detalhados no Apêndice.

Os pinos de conexão da placa do Arduino são projetados para ser acoplados a placas auxiliares denominadas "shields". Entretanto, quando o objetivo é experimental, é possível inserir fios ou terminais (pernas) de cada componente diretamente neles.

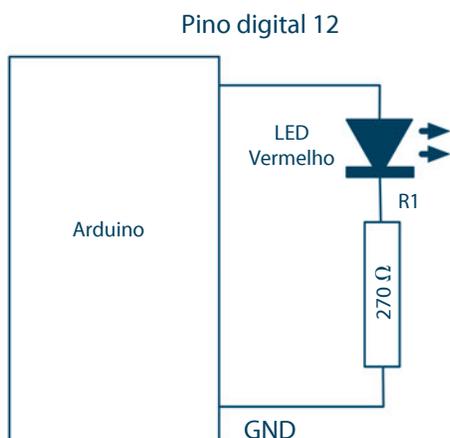
A Figura 1-16 mostra o diagrama esquemático da conexão do LED externo.

Esse tipo de diagrama esquemático usa símbolos especiais para representar os componentes eletrônicos. O LED aparece representado como uma seta grande, indicando que um diodo emissor de luz, como qualquer outro diodo, permite a circulação de corrente apenas em um sentido. As setas pequenas junto ao símbolo do LED indicam que ele emite luz.



**Figura 1-15** Transferindo o sketch para a placa do Arduino.

Fonte: do autor.



**Figura 1-16** Diagrama esquemático de um LED conectado à placa do Arduino.

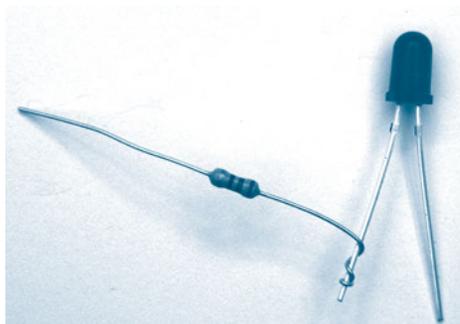
Fonte: do autor.

O resistor está representado simplesmente como um retângulo. Frequentemente os resistores são mostrados na forma de uma linha em zigue-zague. As demais linhas do diagrama representam conexões elétricas entre os componentes. Essas conexões podem ser feitas por fios ou pelas trilhas de uma placa de circuito impresso. Nesse projeto, essas conexões serão os próprios terminais dos componentes.

Podemos conectar os componentes diretamente aos pinos de conexão do Arduino, inserindo-os entre o pino digital 12 e o pino GND (terra), mas primeiro precisamos fazer uma conexão entre um terminal do LED e um do resistor.

Para fazer a conexão com o LED, poderemos usar qualquer um dos terminais do resistor. Entretanto, há um modo correto de ligar o LED. No LED, um terminal é um pouco menor do que o outro. O terminal mais longo deve ser conectado ao pino digital 12, e o mais curto deve ser conectado ao resistor. Nos LEDs e em alguns outros componentes, vale a convenção do terminal positivo ser mais longo do que o negativo.

Para ligar o resistor ao terminal curto do LED, afaste cuidadosamente os terminais do LED e enrosque um terminal do resistor em torno do terminal curto do LED, como está mostrado na Figura 1-17.



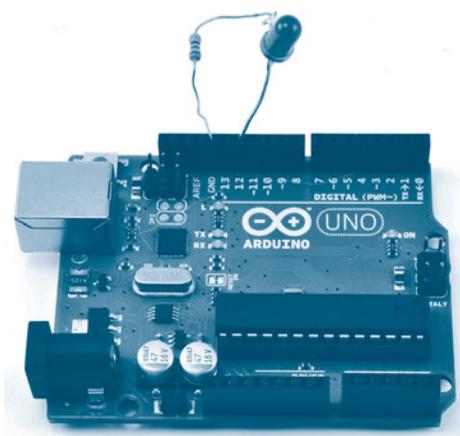
**Figura 1-17** Um LED conectado em série com um resistor.

Fonte: do autor.

A seguir, encaixe o terminal longo do LED no pino digital 12 e o terminal livre do resistor em um dos dois pinos GND de conexão. A Figura 1-18 mostra como fazer isso. Algumas vezes, pode ser útil dobrar ligeiramente a ponta do terminal para que se encaixe mais firmemente no pino de conexão.

Agora podemos modificar o sketch para usarmos o LED externo que acabamos de conectar. Tudo que precisamos fazer é alterar o sketch de modo que o pino digital 12 seja utilizado no lugar do 13 para controlar o LED. Para isso, modificaremos a linha

```
int ledPin = 13;
// LED conectado ao pino digital 13
```



**Figura 1-18** Um LED conectado à placa de Arduino.

Fonte: do autor.

para

```
int ledPin = 12;  
// LED conectado ao pino digital 12
```

Agora, clicando no botão “Upload to IO Board”, transfira o sketch do mesmo modo que você fez quando aumentou a frequência do pisca-pisca.

## » Protoboard

Enroscar fios para fazer conexões só é prático em casos simples, como conectar um LED e um resistor. Um protoboard (ou matriz de contato) permite a construção de circuitos complexos sem necessidade de soldagem. De fato, é uma boa ideia primeiro construir o circuito em um protoboard até que funcione corretamente para, então, realizar as soldagens.

Um protoboard constitui-se de um bloco de plástico com orifícios e com conexões fixadas por molas metálicas. Os componentes eletrônicos são inseridos nesses orifícios, também denominados contatos, pela parte superior.

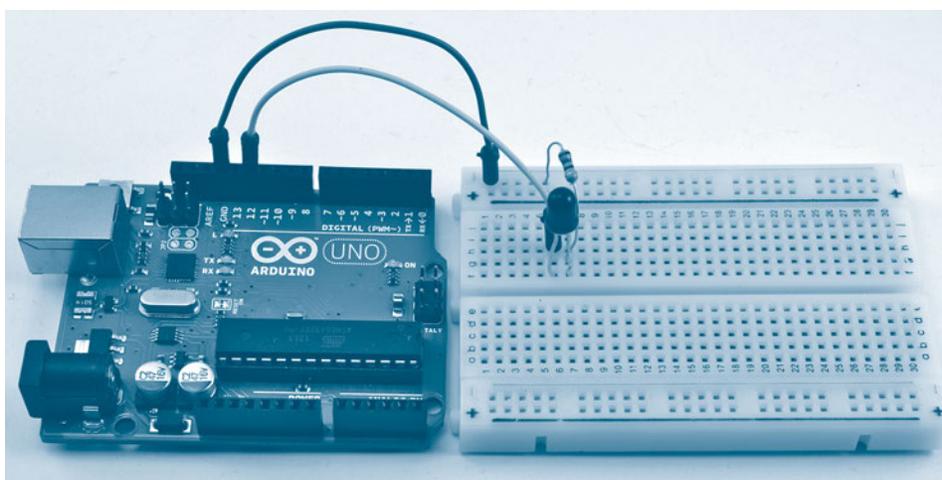
Abaixo dos orifícios do protoboard, há trilhas metálicas que conectam entre si diversos orifícios,

constituindo fileiras de contatos. As fileiras apresentam um intervalo entre elas, de modo que circuitos integrados do tipo DIL (dual-in-line) podem ser inseridos sem que os terminais de uma mesma file sejam colocados em curto-circuito.

Podemos construir esse projeto em um protoboard em vez de enroscar pernas de componentes. A Figura 1-19 mostra uma fotografia de como fazer isso. A Figura 1-20 mostra com mais clareza como os componentes são posicionados e conectados.

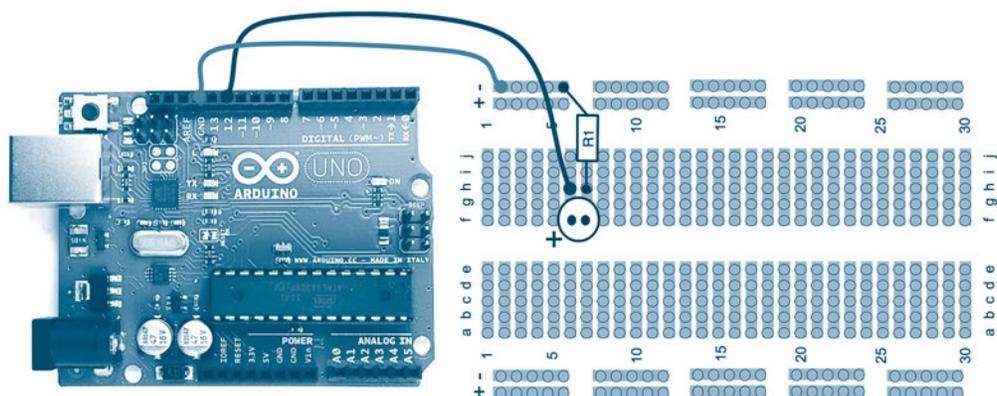
Você observará que, nas bordas do protoboard (em cima e embaixo), há duas faixas horizontais compridas com fileiras de orifícios. Por baixo, há trilhas metálicas de conexão entre os orifícios, estando dispostas perpendicularmente às fileiras normais de conexão. Essas fileiras compridas são usadas para energizar os componentes do protoboard. Normalmente, uma é para o terra (0V ou GND) e uma para a tensão de alimentação positiva (usualmente 5V).

Além do protoboard, você precisará de alguns pedaços de fios (jumpers) para fazer conexões (veja o Apêndice). Costumam ser pedaços curtos de diversas cores e com alguns centímetros de comprimento. São usados para realizar as conexões entre o Arduino e o protoboard. Se quiser, você poderá usar fio rígido e um descascador de fio ou alicate



**Figura 1-19** Projeto 1 montado no protoboard.

Fonte: do autor.



**Figura 1-20** A disposição dos componentes do Projeto 1.

Fonte: do autor.

para descascar as pontas dos fios. É bom dispor de ao menos três cores diferentes: vermelho para as conexões com o lado positivo da alimentação elétrica, preto para o lado negativo e algumas outras cores (amarelo ou laranja) para as demais conexões. Isso facilita muito o entendimento do circuito. Você também poderá comprar fios rígidos de conexão já preparados em diversas cores. Observe que não é aconselhável usar fios flexíveis porque tendem a se dobrar quando são inseridos nos orifícios do protoboard.

Podemos desenroscar os terminais do LED e do resistor, deixando-os retos para, em seguida, inseri-los em um protoboard. O protoboard mais comum à venda tem 60 fileiras de contatos. Em nossos projetos, o protoboard usado frequentemente

terá 30 fileiras de contatos, sendo conhecido como meio protoboard. Cada fileira tem cinco contatos, um intervalo e mais cinco contatos. Neste livro, usaremos seguidamente esse protoboard. Se você conseguir algo similar, facilitará muito. A placa usada foi fornecida pela empresa AdaFruit (veja o Apêndice). O seu tamanho e disposição de contatos são muito comuns.

## »» **Resumo**

Criamos o nosso primeiro projeto, ainda que bem simples. No Capítulo 2, iremos nos aprofundar mais um pouco no Arduino antes de passarmos para projetos mais interessantes.