

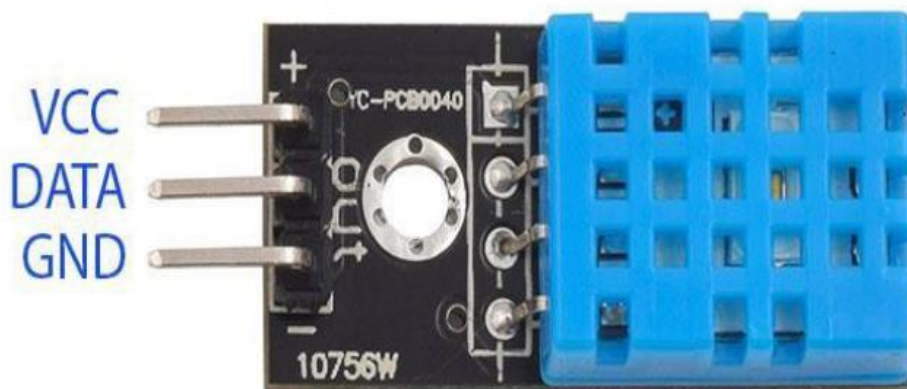
Soy Venture / Grupo 5

1ADSA - Pesquisa e Inovação I 2022/1

| NOME COMPLETO | RA |
|--|----------|
| Caíque Carvalho da Silva | 01221079 |
| Gustavo Lima Aliba | 01221182 |
| Hiago Victor de Lima Rodrigues | 01221192 |
| Kaike Mamede Viana | 01221087 |
| Moisés Matheus Ferreira de Carvalho Oliveira | 01221160 |
| Vittor Gabriel Pavanelli Ribeiro | 01221173 |

DHT11

Após estar no local de instalação, você observará algo semelhante a isso:



Este é o **DHT11**, O Sensor de Umidade e Temperatura DHT11 é um sensor de temperatura e umidade que permite fazer leituras de temperaturas entre 0 a 50 Celsius e umidade entre 20 a 90%, muito usado para projetos com Arduino.

O elemento sensor de temperatura é um termistor(Termístor é um semicondutor sensível à temperatura) do tipo NTC e o sensor de Umidade é do tipo HR202, o circuito interno faz a leitura dos sensores e se comunica a um microcontrolador através de um sinal serial de uma via.

Especificações:

- Modelo: DHT11 (Datasheet)
- Faixa de medição de umidade: 20 a 90% UR
- Faixa de medição de temperatura: 0° a 50°C
- Alimentação: 3-5VDC (5,5VDC máximo)
- Corrente: 200uA a 500mA, em stand by de 100uA a 150uA
- Precisão de umidade de medição: $\pm 5,0\%$ UR
- Precisão de medição de temperatura: ± 2.0 °C
- Tempo de resposta: 2s
- Dimensões: 23 x 12 x 5mm (incluindo terminais)

Jumpers

O que são **Jumpers**: Jumper é um pequeno condutor utilizado para conectar dois pontos de um circuito eletrônico

Aqui nós estamos definindo, o que cada cor dos Jumpers vai representar algo:

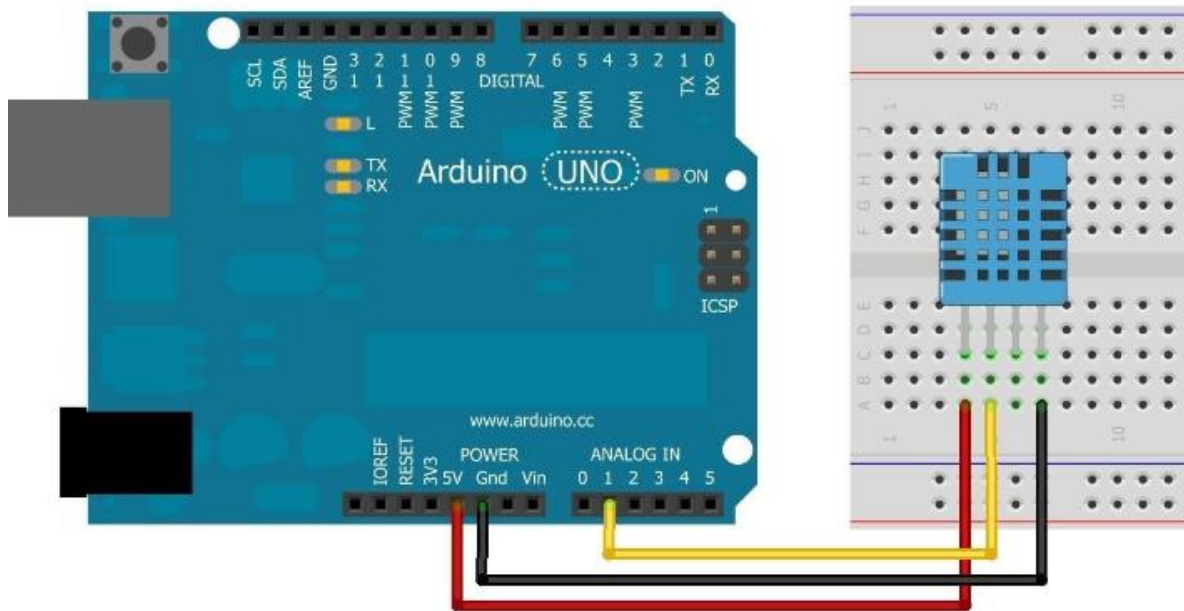
OBS: Não precisa ser necessariamente essas cores, nós só estabelecendo um padrão, para definir o que cada cor faz.

Jumper **vermelho**: 5V (Seria voltagem ou carga utilizada)

Jumper **preto**: GND(Seria a densidade neutra)

Jumper **amarelo**: VCC (Unidade de tensão elétrica, ou seja, tensão em corrente contínua)

Quando conectar na Protoboard, deverá ficar exatamente desse jeito:



ARDUINO IDE

Primeiramente, tenha em mente que você irá precisar de:

- Um computador (Windows, Mac ou Linux)
- Uma placa Arduino ou compatível
- O cabo USB apropriado para a placa escolhida

Instalando a IDE no Windows

Nesta parte iremos mostrar como instalar a IDE nas versões Windows 8, Windows 7, Vista e XP.

Primeiro vá até a página de download da IDE e baixe a versão mais atual para Windows. A IDE não precisará ser instalada, pois é um aplicativo feito em Java e poderá rodar nos mais diversos sistemas operacionais.



ARDUINO 1.6.7

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

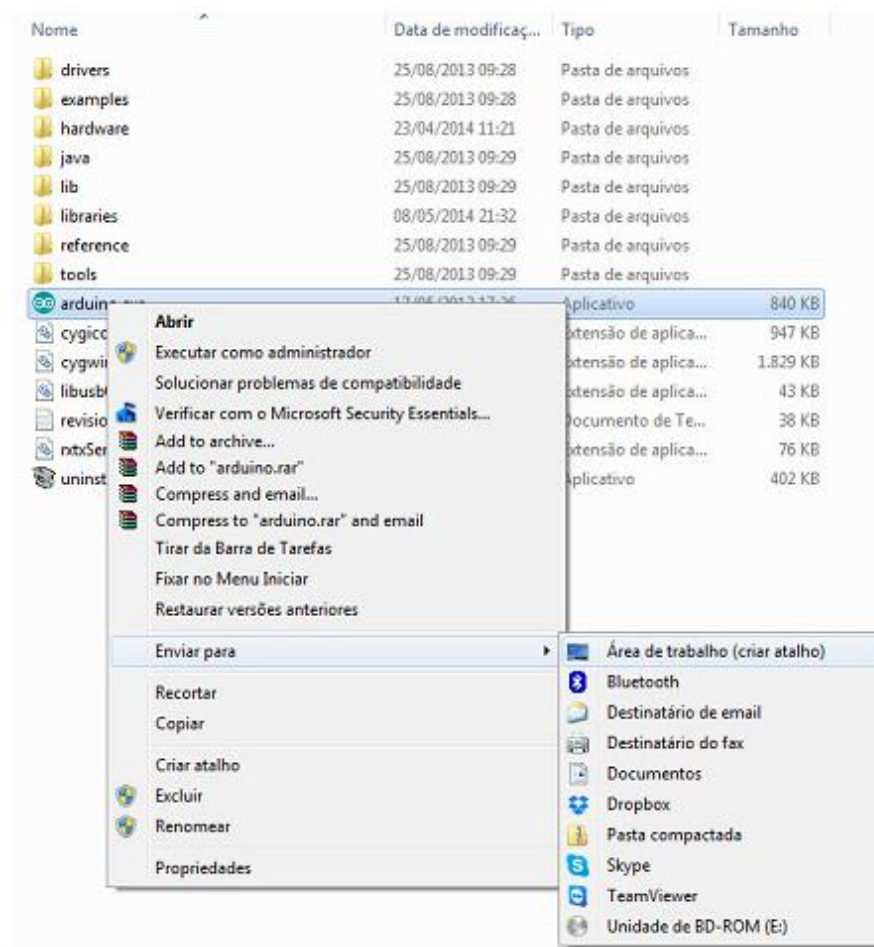
Windows Installer
Windows ZIP file for non admin install

Mac OS X 10.7 Lion or newer

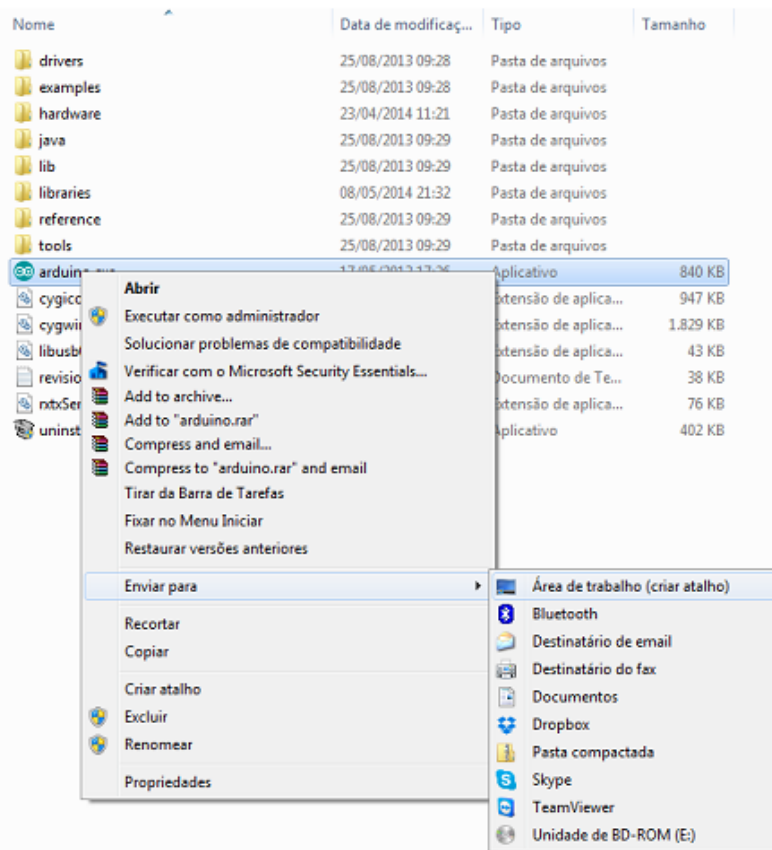
Linux 32 bits
Linux 64 bits

[Release Notes](#)
[Source Code](#)
[Checksums](#)

Quando finalizar o download, descompacte a pasta no diretório: **C:** conforme apresentado na figura abaixo. (É extremamente importante que não sejam alterados os arquivos desta pasta, a menos que você saiba o que está fazendo)



Agora basta criar um atalho da IDE na área de trabalho e você já poderá programar sua placa:



Code:

#include

<DHT.h>

#include <DHT_U.h>

#include <Adafruit_Sensor.h>

#include <DHT.h>

#include <DHT_U.h>

#include "DHT.h"

#define DHTPIN A1

DHT dht(DHTPIN, DHT11);

void setup()

{

```
pinMode(DHTPIN, INPUT);
pinMode(CHAVPIN, INPUT);
Serial.begin(9600);
dht.begin();
}
void loop()
{
float dht11_umidade = dht.readHumidity();
float dht11_temperatura = dht.readTemperature();
Serial.print(dht11_umidade);
Serial.print(";");
Serial.print(dht11_temperatura);
Serial.print(";");
}
```

Funções de alguns comandos: (sla o nome q eu coloco aq)

#include: serve para incluir um arquivo em seu projeto. Pode ser uma biblioteca ou um outro arquivo de código fonte.

#define: permite ao programador dar um nome a um valor constante antes de o programa ser compilado.

#void setup(): A função setup() é chamada quando um sketch inicia. Use-a para inicializar variáveis, configurar o modo dos pinos(INPUT ou OUTPUT).

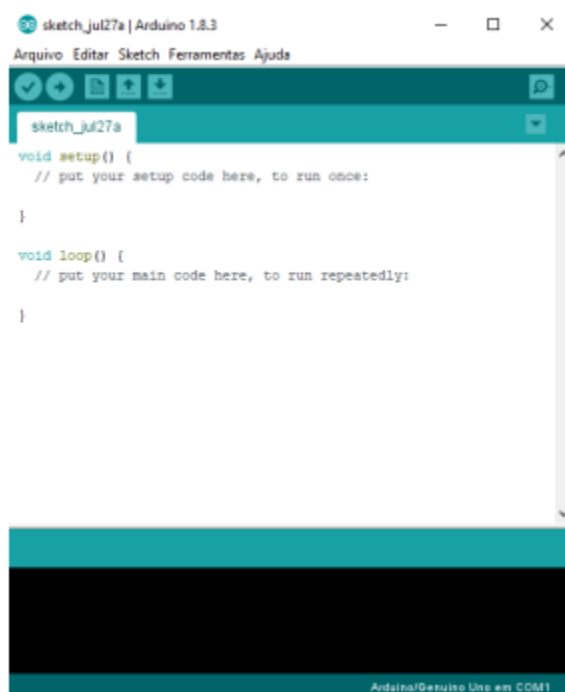
#void loop(): A função loop repete-se consecutivamente as variáveis criadas a partir do void setup(), enquanto a placa estiver ligada.

serial.print(): Imprime dados na porta serial em como texto ASCII (facilmente legível, diferentemente dos valores binários). Essa função pode assumir várias formas. números são impressos usando um caractere ASCII para cada dígito.

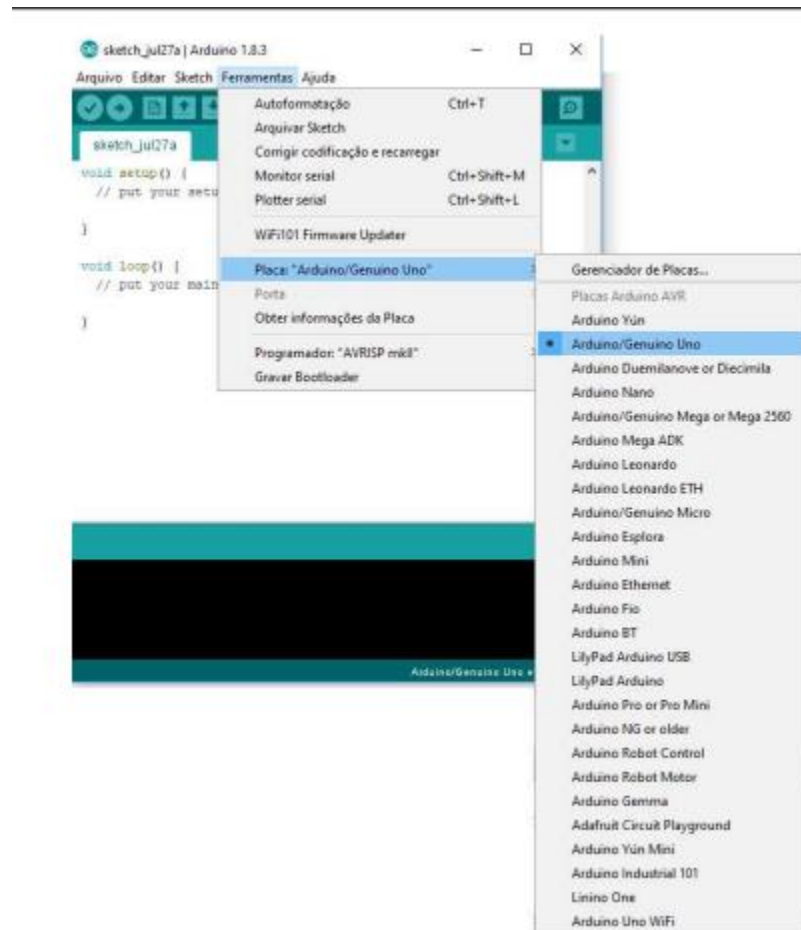
begin() É a primeira função a ser utilizada quando vai trabalhar com a comunicação serial. Ela configura a taxa de comunicação em bits por segundo (baud rate

Configuração

Agora vem a configuração, que é uma etapa importante. Primeiramente, clique no ícone do Arduino que foi criado na área de trabalho. Se o Firewall do Windows bloquear algum recurso, será mostrada uma alerta de segurança. Nesse caso, clique em Permitir acesso. Finalmente, aparecerá uma janela com a IDE do Arduino. Muito provavelmente o software estará no idioma do seu Windows. Observe que o sketch já vem com um esqueleto de um código a ser criado, com as funções `setup` e `loop`. Falaremos sobre as funções na linguagem do Arduino na seção 5.9.

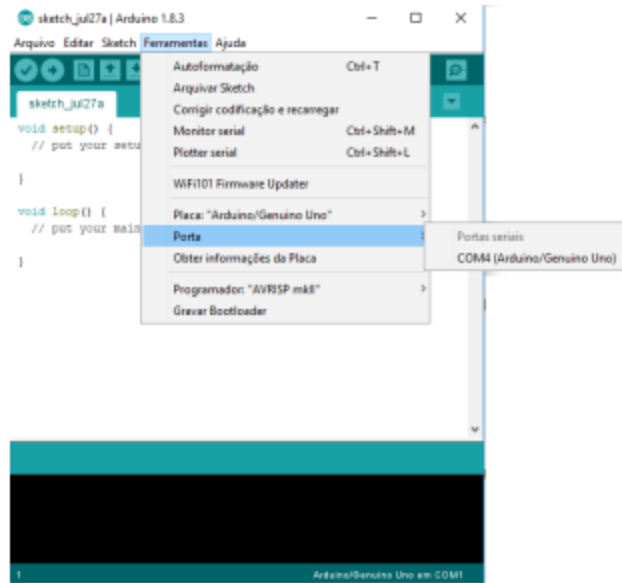


Agora você deve escolher a placa do Arduino que você vai trabalhar. Para isso, clique em Ferramentas --> Placa: "Arduino/Genuíno Uno e faça a escolha apropriada.



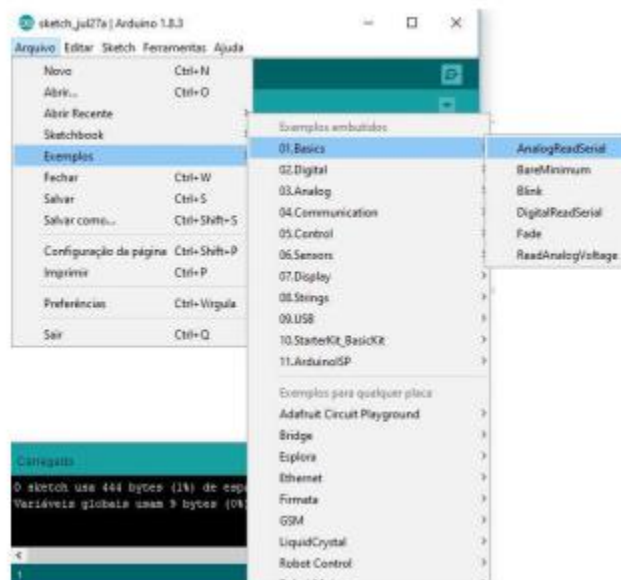
Escolhendo a placa a ser usada.

A próxima escolha diz respeito a porta USB do computador. Para isso, plugue o cabo USB da sua placa em uma porta USB do computador. Em seguida, clique em Ferramentas --> Porta e selecione uma das portas disponíveis.



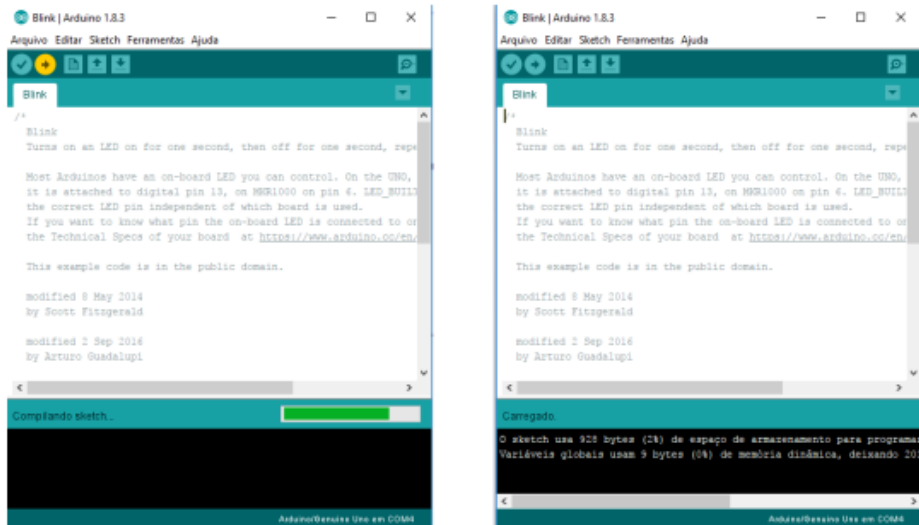
Escolhendo a porta a ser usada

Por fim, vamos testar a Comunicação entre o Arduino e o computador usando um código já pronto. Clique em Arquivo → Exemplos → 01.Basics → Blink.



Abrindo o código Blink.

Em seguida clique em Carregar. Se tudo der certo, será possível ver uma pequena luz amarela acendendo e apagando na placa do Arduino.



Do lado esquerdo, temos o código sendo carregado. Do lado direito vemos o código carregado com sucesso.