



UFAC



FUNDAPE



CITS

**Internet das Coisas (IoT)
para a Indústria 4.0**



PROJETO IOT



Introdução a IoT

Internet das Coisas

Prof. André Nasserla
andre.nasserla@ufac.br

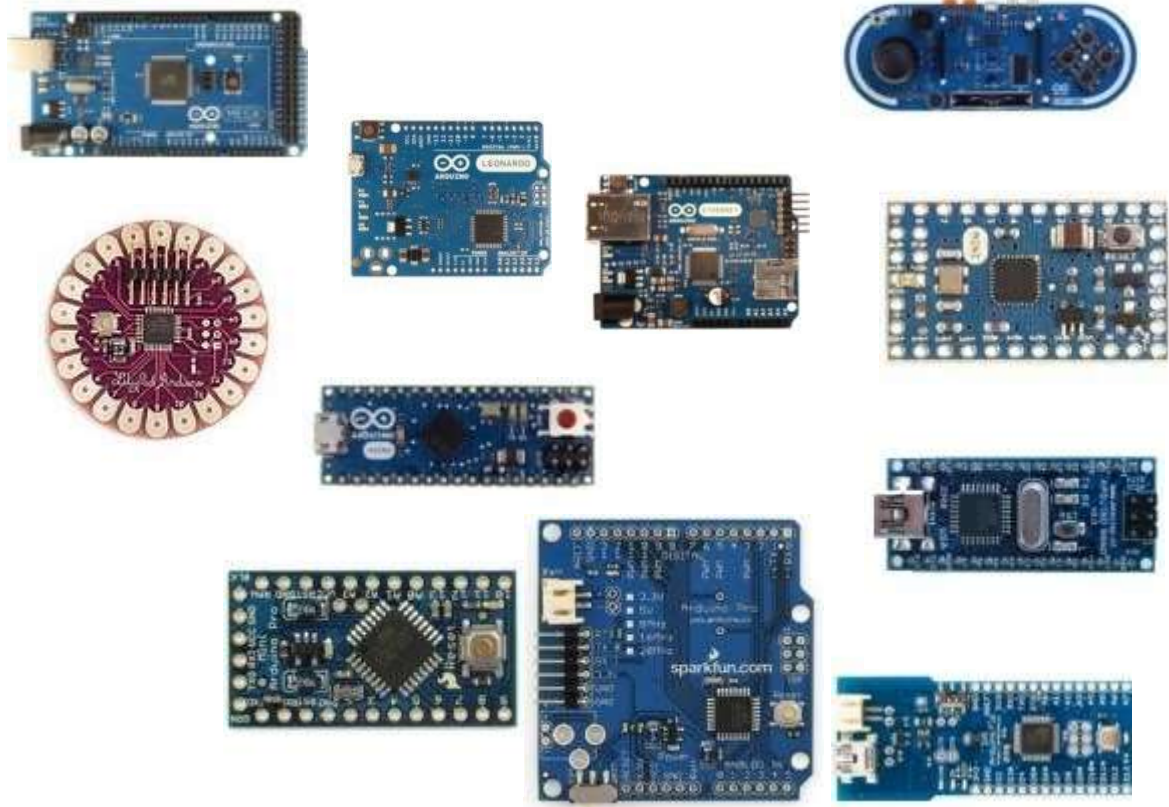
Apresentação

- O Arduino é uma plataforma utilizada para prototipação de circuitos eletrônicos.
- O projeto do Arduino teve início em 2005 na cidade de Ivrea, Itália.
- O Arduino é composto por uma placa com microcontrolador Atmel AVR e um ambiente de programação baseado em C++.
- Tanto o hardware como o ambiente de programação do Arduino são livres, ou seja, qualquer pessoa pode modificá-los e reproduzi-los.
- O Arduino também é conhecido como plataforma de computação física.

Apresentação

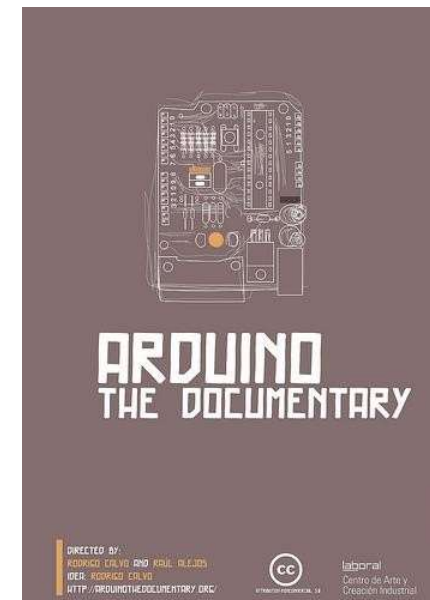
- Tipos de Arduino
- Existem vários tipos de Arduino com especificidades de hardware. O site oficial do Arduino lista os seguintes tipos:

- Arduino UNO
- Arduino Leonardo
- Arduino Due
- Arduino Esplora
- Arduino Mega
- Arduino Ethernet
- Arduino Mini
- Arduino LilyPad
- Arduino Micro
- Arduino Nano
- Arduino Pro
- Arduino Fio



Apresentação

- Referências na WEB
- O site oficial do Arduino é <http://arduino.cc>
 - Um documentário sobre o Arduino pode ser assistido em: <http://arduinothedocumentary.org/>



Arduino na automação

- Um dos projetos mais cobijados pelos amantes da tecnologia é a automação residencial com Arduino, onde, por meio de aplicativos, comandar a casa como um todo, permitindo abrir e fechar cortinas e janelas motorizadas, ligar e desligar televisores em horários pré-definidos, comandar ventiladores, e tudo mais, isso diretamente do celular, tablet ou computador.



Arduino na automação

- Grandes coisas em automação podem ser feitas adicionando ao Arduino um **Ethernet Shield**, permitindo transformar o pequeno microcontrolador em um dispositivo conectado à internet, capaz de mudar o estado de luzes, TVs, praticamente qualquer coisa que pode pensar, ligado e desligado, utilizando uma interface baseada em browser ou um temporizador.



Uso e aplicação

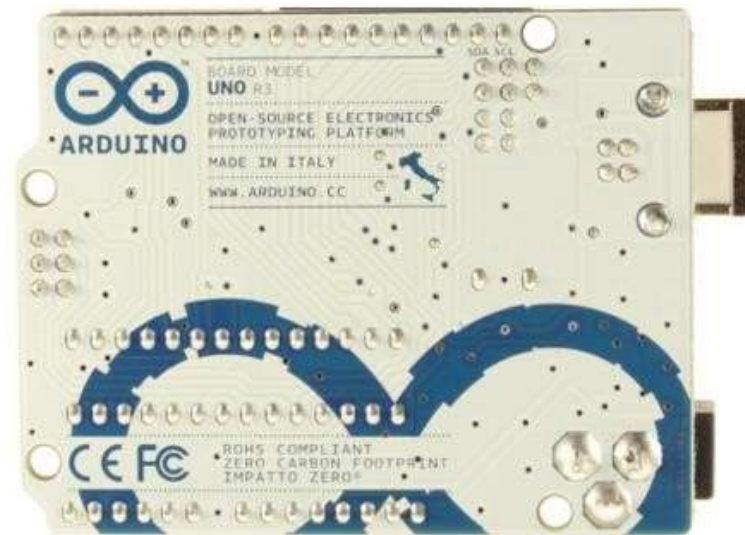
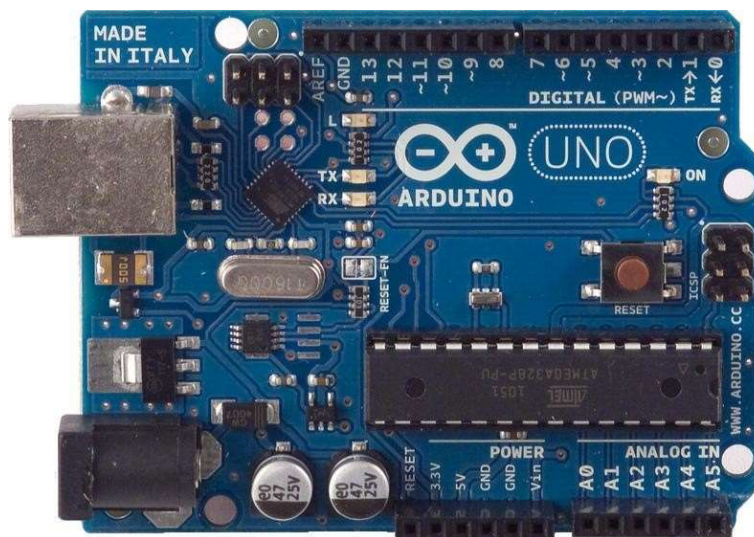
- A plataforma Arduino foi criada em 2005 como uma grande alternativa para o ensino e aprendizagem de microcontroladores.
- De fácil utilização, é considerada uma grande revolução na popularização da programação e no universo da IOT (internet das coisas).
- É interessante aprender a utilizar essa plataforma, pois o mercado de trabalho está a cada dia mais interessado em profissionais com capacidade de compreensão do funcionamento de dispositivos automatizados e interligados entre si.
- Com a plataforma Arduino é possível criar projetos completos de automação e produtos de fácil utilização e baixo custo.
- Isso dará a possibilidade de pesquisar e criar novos dispositivos para estudo ou comercialização.

Como Começar?

- Para o melhor se encaixar nesse mercado, bem como compreender melhor as necessidades das aplicações, alguns conhecimentos são interessantes:
- 1 – Básico de eletrônica.
- 2 – Programação em C e C++ ou Python.
- 3 – Entender o funcionamento de microcontroladores:
 - Arduíno;
 - Esp32;
 - RP-Pico e outros;

Arduino UNO

- Vista da placa do Arduino UNO Rev 3 (frente e verso)



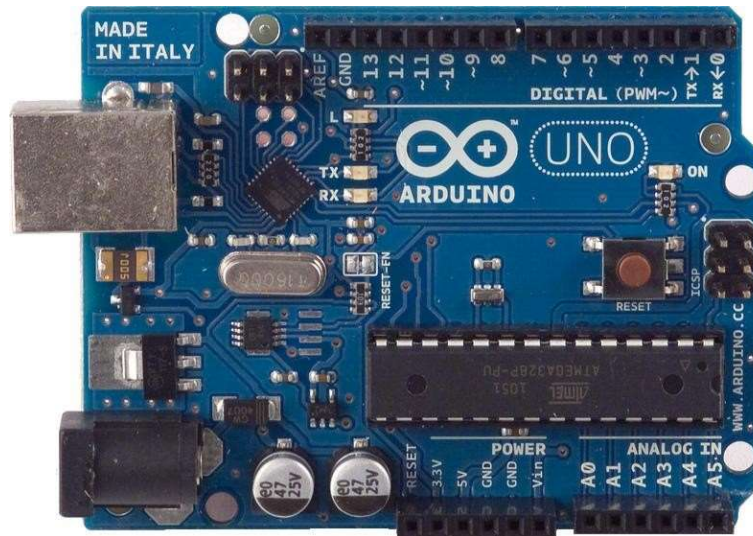
Arduino UNO

- Características

- Microcontrolador: **ATmega328**
- Tensão de operação: **5V**
- Tensão recomendada (entrada): **7–12V**
- Limite da tensão de entrada: **6–20V**
- Pinos digitais: **14 (seis pinos com saída PWM)**
- Entrada analógica: **6 pinos**
- Corrente contínua por pino de entrada e saída: **40 mA**
- Corrente para o pino de 3.3 V: **50 mA**
- Quantidade de memória FLASH: **32 KB (ATmega328) onde 0.5 KB usado para o bootloader**
- Quantidade de memória SRAM: **2 KB (ATmega328)**
- Quantidade de memória EEPROM: **1 KB (ATmega328)**
- Velocidade de clock: **16 MHz**

Arduino UNO

- Alimentação
 - O **Arduino UNO** pode ser alimentado pela porta USB ou por uma **fonte externa DC**.
 - A recomendação é que a **fonte externa seja de 7 V a 12 V** e pode ser ligada diretamente no conector de fonte ou nos pinos **Vin** e **Gnd**.



Arduino UNO

- Pinos



Pinos 3V3, 5V e Gnd: dos 6 pinos dessa barra somente os quatro do meio servem para alimentar um circuito externo conectado ao Arduino: o pino de 5V e o terra (os dois pinos Gnd entre 5V e Vin); e o pino 3V3 que disponibiliza essa tensão com uma corrente máxima de 50mA.

Arduino UNO

- Pinos



Pinos 0 a 5: esses 6 pinos aceitam tensões entre zero e 5 volts CC que vão ao conversor A/D de 10 bits no microcontrolador do Arduino. O pino AREF, numa das barras de pinos digitais, é a entrada de tensão de referência para esse conversor.

Arduino UNO

- Pinos

Pinos AREF e GND: o pino AREF é a entrada de tensão de referência para o conversor A/D do Arduino; o pino GND é o terra, comum a todos os outros pinos.

Pinos 3, 5 e 6 e 9 a 11 (PWM): 6 pinos dos 14 pinos digitais podem ser usados para gerar sinais analógicos com a função `analogWrite()` utilizando a técnica de Modulação por Largura de Pulso (PWM).

Pinos 0 e 1: os dois primeiros pinos digitais são conectados a USART do microcontrolador do Arduino para comunicação serial com um computador.

Pinos 2 e 3: pinos que chamam uma ISR (Interrupt Service Routine) para tratar uma interrupção com a função `attachInterrupt()` nesses pinos.

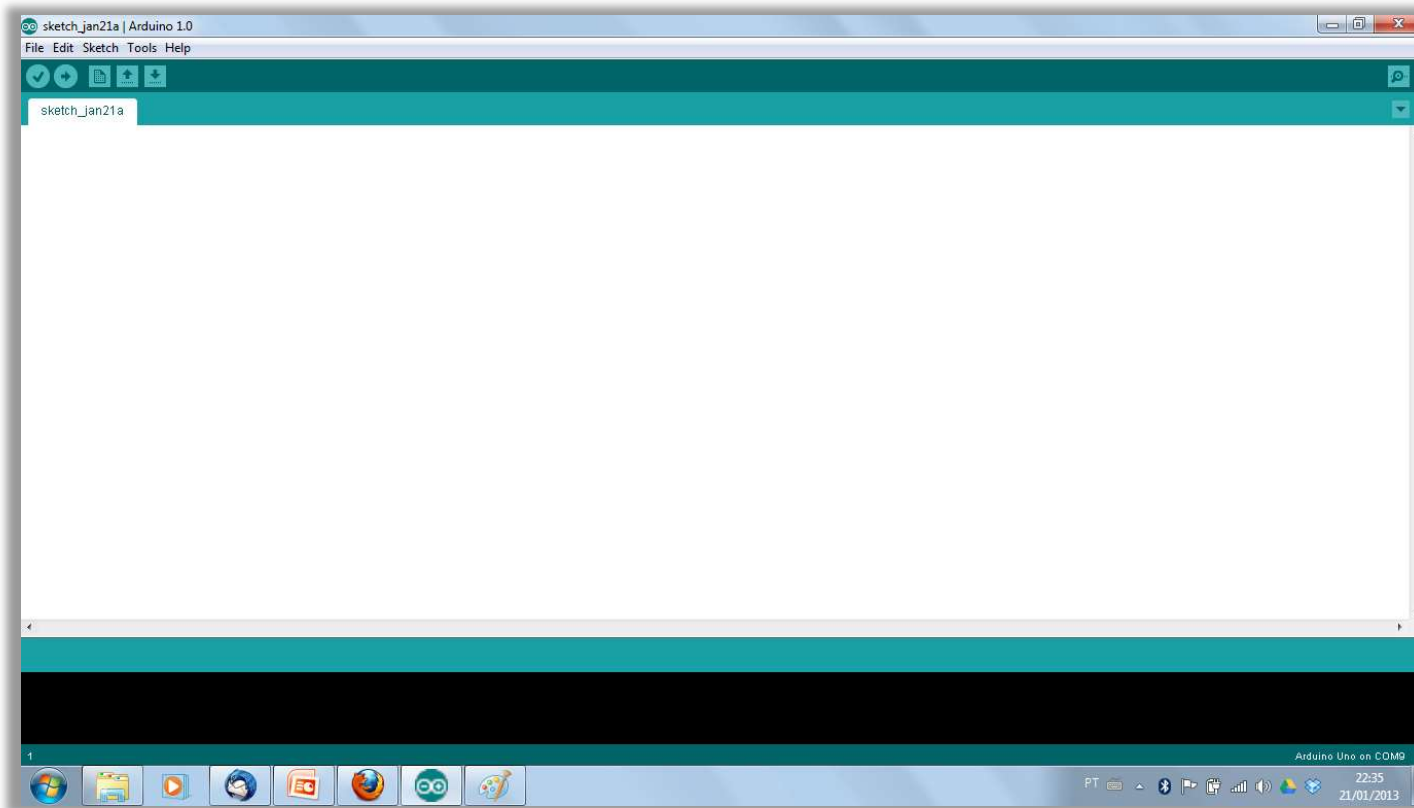


Ambiente de Desenvolvimento

- O ambiente de desenvolvimento do Arduino (IDE) é gratuito e pode ser baixado no seguinte endereço: arduino.cc.
- As principais funcionalidades do IDE do Arduino são:
 - Escrever o código do programa;
 - Salvar o código do programa;
 - Compilar um programa;
 - Transportar o código compilado para a placa do Arduino.

Ambiente de Desenvolvimento

- Interface principal do ambiente de desenvolvimento:



Funções setup() e loop()

- Primeiro programa: Blink LED

```
blink_led  
  
void setup()  
{  
    pinMode(13, OUTPUT);  
}  
  
void loop()  
{  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

Funções `setup()` e `loop()`

- As duas principais partes (funções) de um programa desenvolvido para o Arduino são:
- **setup()**: onde devem ser definidas algumas configurações iniciais do programa. Executa uma única vez.
- **loop()**: função principal do programa. Fica executando indefinidamente.
- Todo programa para o Arduino deve ter estas duas funções.

Funções setup() e loop()

- Formato das funções setup() e loop()

```
setuploop  
void setup()  
{  
  
}  
  
void loop()  
{  
  
}
```

Constantes

- A criação de constantes no Arduino pode ser feita de duas maneiras:
- Usando a palavra reservada `const`
- Exemplo:
- `const int x = 100;`
- Usando a palavra reservada `define`
- Exemplo:
- `#define X 100;`

Constantes

- No Arduino existem algumas constantes previamente definidas e são consideradas palavras reservadas.
- As constantes definidas são:
- **true** – indica valor lógico verdadeiro.
- **false** – indica valor lógico falso.
- **HIGH** – indica que uma porta está ativada, ou seja, está em 5V.
- **LOW** – indica que uma porta está desativada, ou seja, está em 0V.
- **INPUT** – indica que uma porta será de entrada de dados.
- **OUTPUT** – indica que uma porta será de saída de dados.

Comentários

- Muitas vezes é importante comentar alguma parte do código do programa.
- Existem duas maneiras de adicionar comentários a um programa em Arduino.
- A primeira é usando `//`, como no exemplo abaixo:
- `//` Este é um comentário de linha
- A segunda é usando `/* */`, como no exemplo abaixo:
- `/*` Este é um comentário de bloco. Permite acrescentar comentários com mais de uma linha `*/`

Comentários

- Primeiro programa comentado

```
comentarios
/*****
*      OFICINA DE ROBÓTICA - LARM - UFSC      *
*                                              *
* Blink Led: Primeiro programa em Arduino.    *
*      Pisca um led conectado à porta 13.    *
*****/

// função usada para configurações iniciais
void setup()
{
    pinMode(13, OUTPUT);
}

// principal função do programa - laço infinito
void loop()
{
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);
}
```


Portas Digitais

- O Arduino possui tanto portas digitais quanto portas analógicas.
- As portas servem para comunicação entre o Arduino e dispositivos externos, por exemplo: ler um botão, acender um led ou uma lâmpada.
- Conforme já mencionado, o Arduino UNO, possui 14 portas digitais e 6 portas analógicas (que também podem ser utilizadas como portas digitais).

Portas Digitais

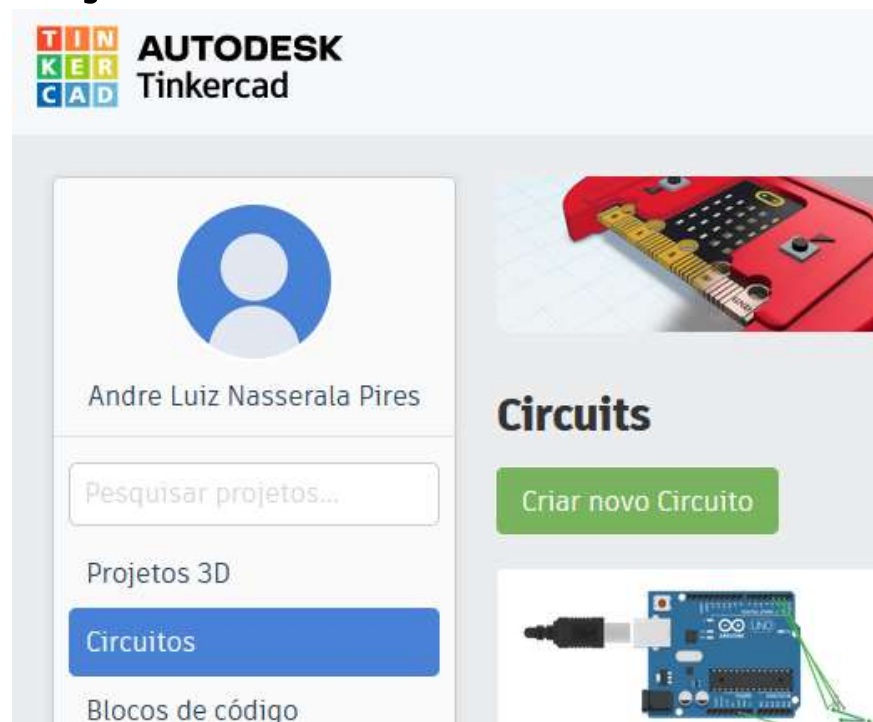
- As portas digitais trabalham com valores bem definidos. No caso do Arduino esses valores são 0V e 5V.
- 0V indica a ausência de um sinal e 5V indica a presença de um sinal.
- Para escrever em uma porta digital basta utilizar a função **digitalWrite(pino, estado)**.
- Para ler um valor em uma porta digital basta utilizar a função **digitalRead(pino)**.

Portas Digitais

- As portas digitais são usadas para entrada e saída de dados.
- Para definir se uma porta será usada para entrada ou para saída de dados, é necessário explicitar essa situação no programa.
- A função `pinMode(pino, estado)` é utilizada para definir se a porta será de entrada ou saída de dados.
- **Exemplos:**
 - Define que a porta 13 será de saída
 - `pinMode(13, OUTPUT)`
 - Define que a porta 7 será de entrada
 - `pinMode(7, INPUT)`

Primeiro Programa

- Usaremos o simulador:
- <https://www.tinkercad.com/>
- Entre e façam uma conta de aluno.



Primeiro Programa

- void setup()
- {
- pinMode(2, OUTPUT);
- }

- void loop()
- {
- digitalWrite(2, HIGH);
- delay(1000);
- digitalWrite(2, LOW);
- delay(1000);
- }

Referências:

- MONK, Simon. Programação com Arduino. Porto Alegre – RS. Editora: Bookman – 2017. ISBN: 9788582604465
- VIDAL, Vitor, Gustavo Murta. Arduino Start. Eletrogate – 2018. Belo Horizonte – MG. Disponível em: <https://conteudo.eletrogate.com/apostila-arduino-start>.
- MALVINO, Albert Paul. Eletrônica: Volume 1. 4.ed. São Paulo – SP: Makron Books, 1997. ISBN: 8534603782.
- SENAI, Senai SP. FUNDAMENTOS DE ELETRÔNICA - 1ªED. Editora: Senai SP – São Paulo 2015. ISBN: 9788583932086
- WILSON, J. A. e Milton Kaufman. Eletrônica Básica - Teoria e Prática - Volume 2. São Paulo: Editora: Rideel, 1980.
- PEREZ, Anderson Luiz Fernandes, Heron Pereira, Cristiano Pereira de Abreu, Renan Rocha Darós. Oficina de Robótica. UFSC – Programação Básica em Arduino - 2015. Disponível em: <http://oficinaderobotica.ufsc.br/programacao-basica-em-arduino/>.