

MBA<sup>+</sup>

APICON *HANDS-on LAB*  
INTERNET OF THINGS

# José Castillo Lema

Formado em Engenharia da Computação pela Universidade da Coruña (Espanha). Mestre em Sistemas e Computação pelo Programa Pós-Graduação em Sistemas e Computação (PPGSC) na linha de pesquisa de Sistemas Integrados e Distribuídos. Hoje em dia pesquisador e doutorando no Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) da Escola Politécnica da Universidade de São Paulo (Poli/USP) na linha de pesquisa de Sistemas Eletrônicos. Autor e co-autor de vários artigos publicados em anais de conferências nacionais/internacionais e revistas na área de redes de computadores e telecomunicações. Participa de vários projetos de pesquisa financiados por agências nacionais e internacionais. Seus principais interesses incluem internet do futuro, redes definidas por software, computação em nuvem, funções de rede virtualizadas, internet das coisas e redes orientadas à conteúdo.



- 1960 – *Mainframe computing*
- 1980 – *Personal computing*
- 2000 – *Internet computing*
- 2010 – *Mobile/Cloud computing*
- 2020 – ***Internet of Things***

## Internet-connected things

**20.8 billion<sup>1</sup>**  
(predicted)

20 ◀ Numbers in billions

19

### The insecurity of things

18

**Medical devices.** Researchers have found potentially deadly vulnerabilities in dozens of devices such as insulin pumps and implantable defibrillators.

16

**Smart TVs.** Hundreds of millions of Internet-connected TVs are potentially vulnerable to click fraud, botnets, data theft and even ransomware, according to Symantec research.

14

**Cars.** Fiat Chrysler recalled 1.4 million vehicles after researchers demonstrated a proof-of-concept attack where they managed to take control of the vehicle remotely. In the UK, thieves hacked keyless entry systems to steal cars.

13

12

11

Today in the USA, there are  
**25 connected  
devices per  
100 inhabitants<sup>1</sup>**

10

9

8

7

6

5

4

3

2

1

**6.4 billion****4.9 billion****3.9 billion**

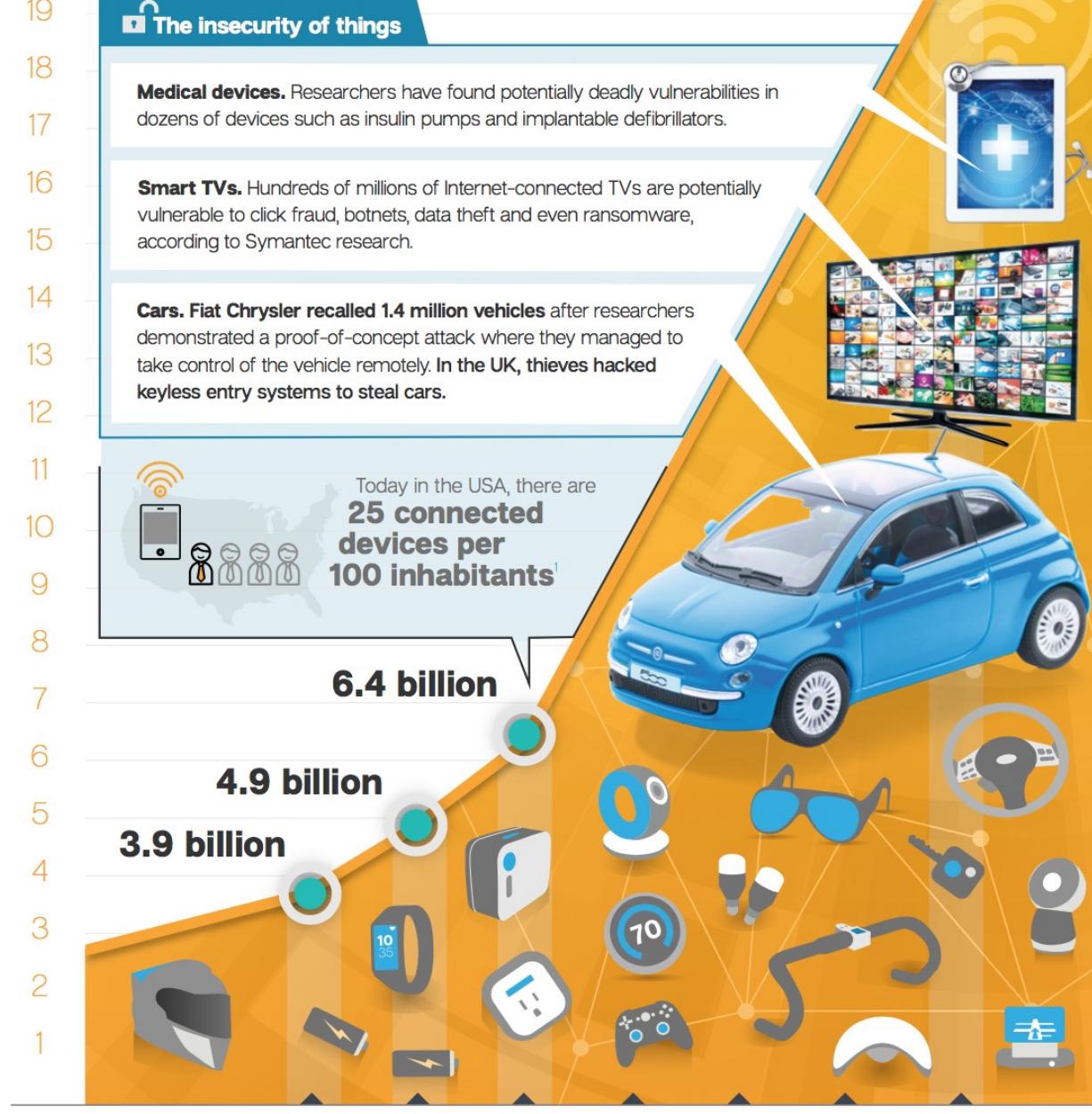
2014

2015

2016

2020

<sup>1</sup> Source: gartner.com/newsroom/id/3165317



# O que é IoT?

A **Internet das Coisas** vem recebendo grande atenção da comunidade **acadêmica** e da **indústria de alta tecnologia**, integrada com os recursos oferecidos pela **computação em nuvem** potencializa o surgimento de estruturas maiores ao exemplo das **cidades inteligentes**.



Coneectar o mundo

Muitas definições,  
mas as principais **características** são:

- Coisa → não um computador/telefone/tablet
- Interface entre o mundo real-virtual
- Altamente conectado
- Inteligente – *smart*
- Grande quantidade de dispositivos heterogêneos

Soluções **conectadas** que combinam **sensores** e **inteligência** para permitir que objetos e infraestrutura interajam com sistemas de **monitoramento, análise** e **controle através de redes** de Internet.

Fonte: Forrester



# Por que IoT?

# | É impulsionado por redução de custo

FIAP

- **Largura de banda** barateou em **40x**
  - Custo por **sensor** passou de **\$1.30** para **\$0.60**
  - Custo de **processamento** diminuiu em **60x**
- 
- Maior **cobertura** das redes de dados
  - Novas tecnologias de **rede** voltadas ao IoT
  - Vida útil estendida das **baterias**

# | Por que desenvolver soluções IoT?

FIAP

**Para coletar dados:** O acesso à Internet já se encontra altamente monitorado para caracterizar o comportamento dos usuários.

Mas não o mundo real, excetuando o GPS.  
O conhecimento dos comportamentos das pessoas no mundo real, seus movimentos, costumes,... é muito valioso.

De forma análoga aos dispositivos móveis, muitos dispositivos IoT coletam dados dos usuários

# | Por que desenvolver soluções IoT?

FIAP

**Para otimizar:** um sistema de ar condicionado automatizado pode melhorar a qualidade de vida das pessoas.

Mas também pode permitir que a concessionária de energia limite o consumo de energia durante determinados dias.

Uma eletricidade 20% mais barata durante todo o ano em troca de permitir a companhia elétrica limitar em 10% o consumo durante 3 dias por ano?

... por que não? :-)

# Dispositivos IoT

# | De vestir

FIAP

Coisas **de vestir**, interação  
frequentemente conectadas ao telefone.

Motivação: informações corpo/saúde.



Withings JAWBONE BASIS

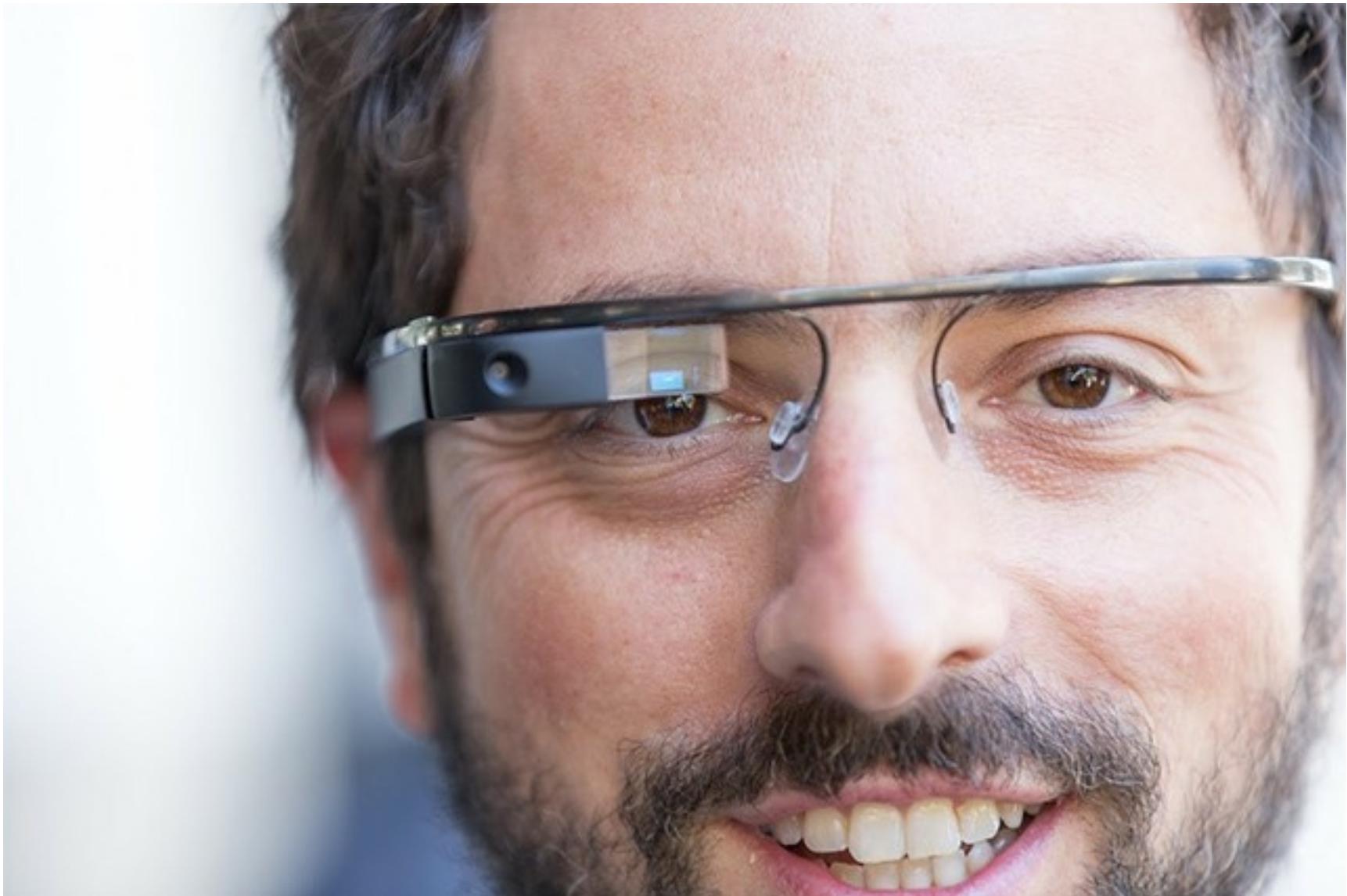
fitbit.



**Relógios...** mas não necessariamente  
Mais similar a um dispositivo móvel:

- Colaboração Intel/TagHeuer/Google
- Pulseira inteligente Intel MICA
- Apple Watch

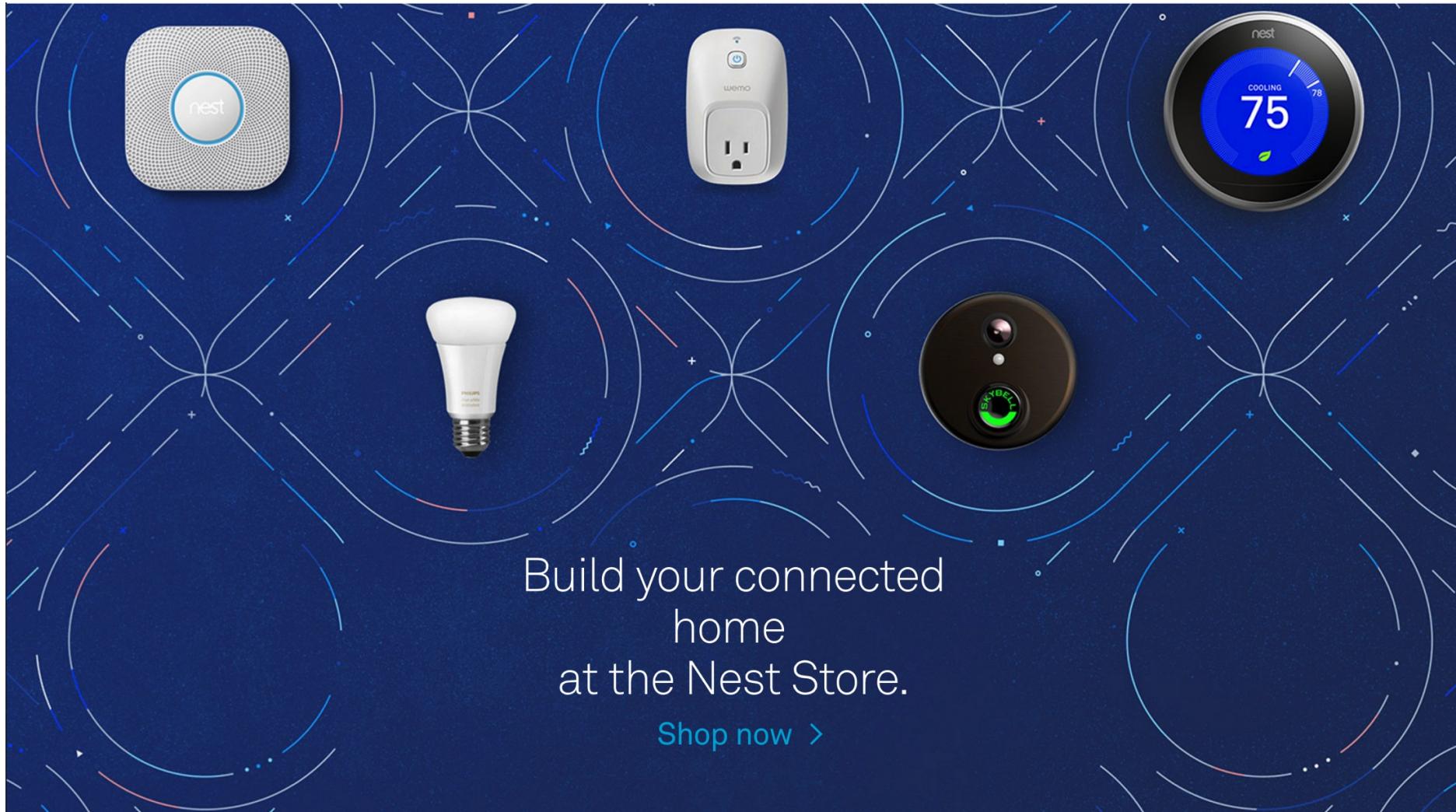




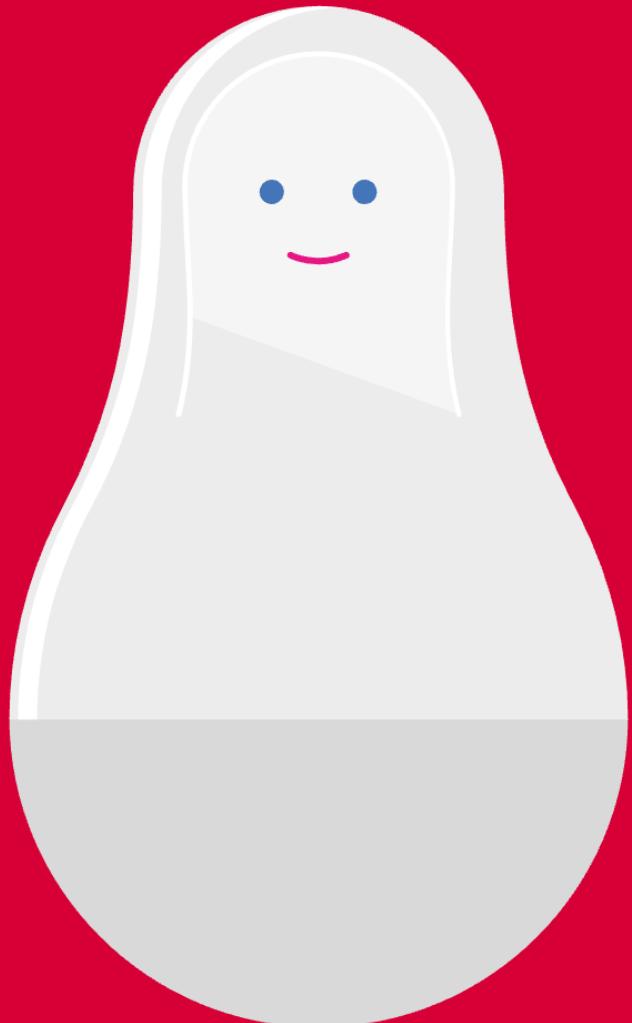
# Edifícios inteligentes – Automação

FIAP

## Google Nest



Sen.se mother



mother.

The one simple way to:  
**optimize home  
heating**

# | Robôs e drones

FIAP



## Sinalização digital / Mobiliário inteligente



## Superfícies inteligentes



MASH MACHINE

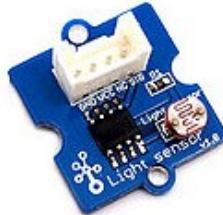
# Sensores/Atuadores IoT

# Sensores para monitoramento ambiental

FIAP



Grove - Digital Light Sensor



Grove - Light Sensor



Grove - Temperature and  
Humidity Sensor



Grove - Barometer Sensor



Grove - Dust Sensor



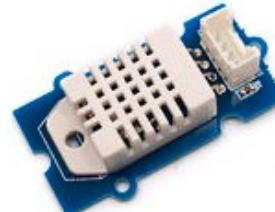
Grove - Gas Sensor



Grove - Temperature Sensor



Grove - Air Quality Sensor



Grove - Temperature and  
Humidity Sensor Pro



Grove - Gas Sensor(O<sub>2</sub>)

# Sensores de movimento



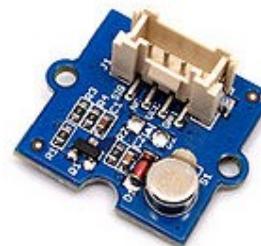
Grove - 3-Axis Digital Compass



Grove - 3-Axis Digital Accelerometer( $\pm 1.5g$ )



Grove - 3-Axis Digital Gyro



Grove - Collision Sensor



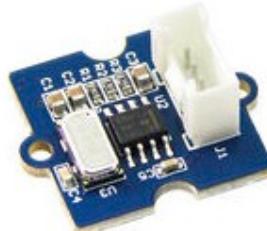
Grove - 3-Axis Analog Accelerometer



Grove - 3-Axis Digital Accelerometer( $\pm 16g$ )



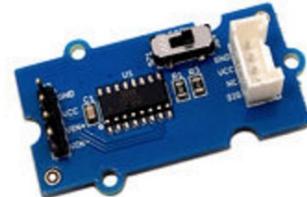
Grove - 6-Axis Accelerometer and Compass V1.0



Grove - Single Axis Analog Gyro

# Sensores para monitoramento físico

FIAP



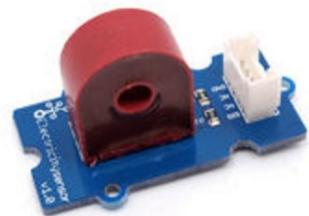
Grove - Water Sensor

Grove - Magnetic Switch

Grove - Alcohol Sensor

Grove - RTC

Grove - Differential Amplifier



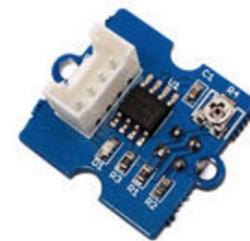
Grove - Electricity Sensor

Grove - Sound Sensor

Grove - IR Distance Interrupt

Grove - Tilt Switch

Grove - Encoder



Grove - I2C Color Sensor

Grove - Sound Recorder

Grove - Moisture Sensor

Grove - PIR Motion Sensor

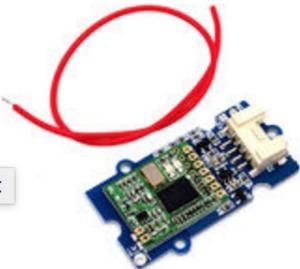
Grove - Infrared Reflective Sensor

# Comunicação sem fio

FIAP



Grove - 315MHz Simple RF Link Kit



Grove - Serial RF Pro



Grove - GPS



Grove - 125KHz RFID Reader



Grove - Serial Bluetooth



Grove - 433MHz Simple RF Link Kit



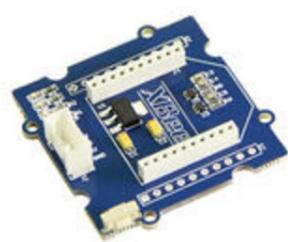
Grove - XBee Carrier



Grove - Infrared Receiver



Grove - Infrared Emitter



Grove - Bee Socket

# Interface com o usuário



Grove - Solid State Relay



Grove - OLED Display 128\*64



Grove - Serial LCD



Grove - LED Socket Kit



Grove - Button



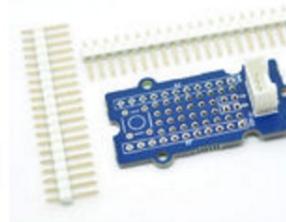
Grove - Vibration Motor



Grove - LED Bar



Grove - Relay



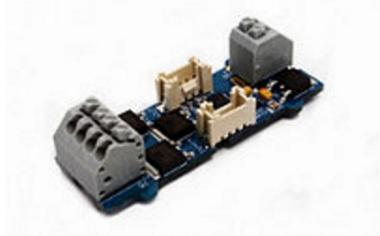
Grove - Protoshield



Grove - Thumb Joystick



Grove - Infrared Reflective Sensor



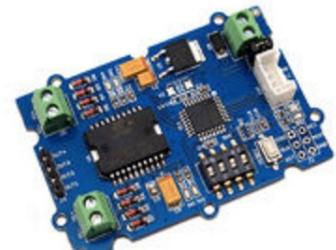
Grove - LED Strip Driver



Grove - 4-Digit Display



Grove - OLED Display 96\*96

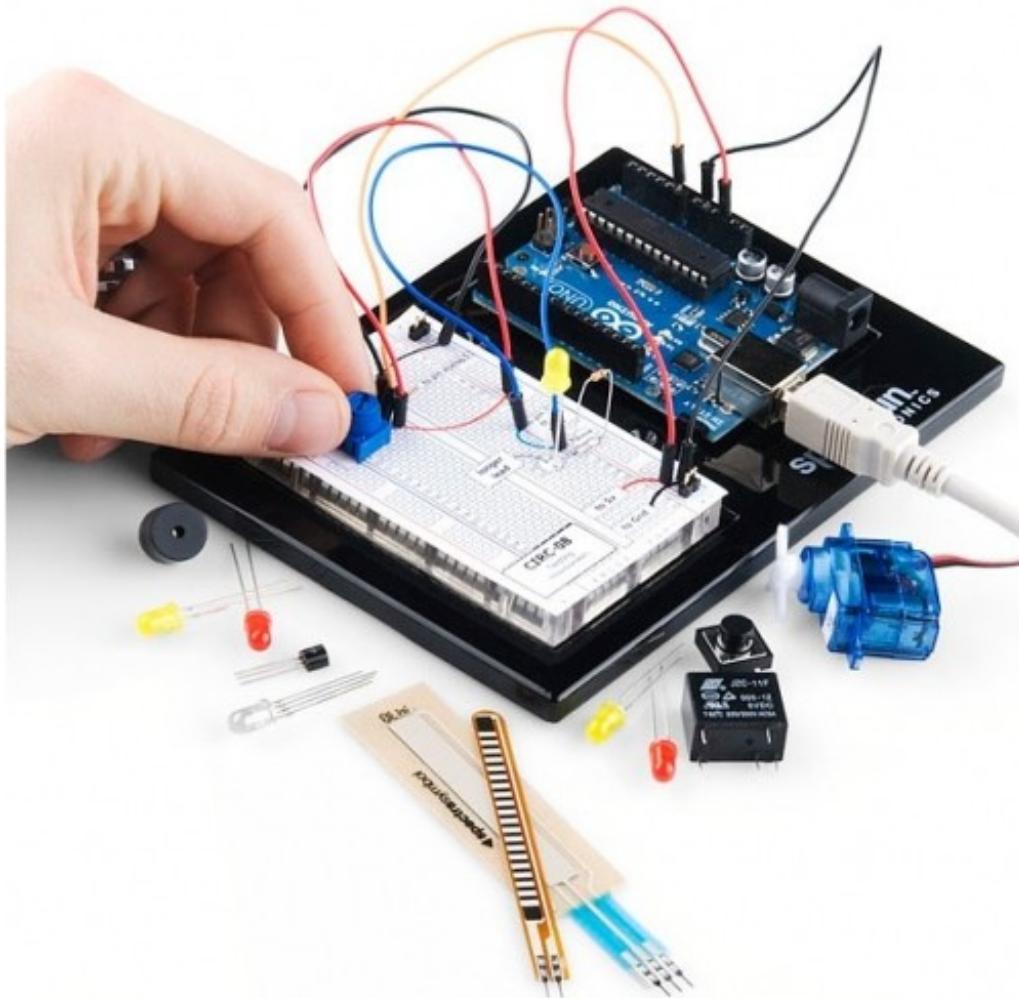


Grove - I2C Motor Driver

# Placas IoT

# Arduino Uno

FIAP



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.0". The menu bar includes File, Edit, Sketch, Tools, and Help. A toolbar with various icons is at the top. The main area displays the "Blink" sketch:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);      // set the LED on
  delay(1000);                // wait for a second
  digitalWrite(13, LOW);       // set the LED off
  delay(1000);                // wait for a second
}
```

The status bar at the bottom indicates "1" and "Arduino Uno on /dev/ttyACM1".

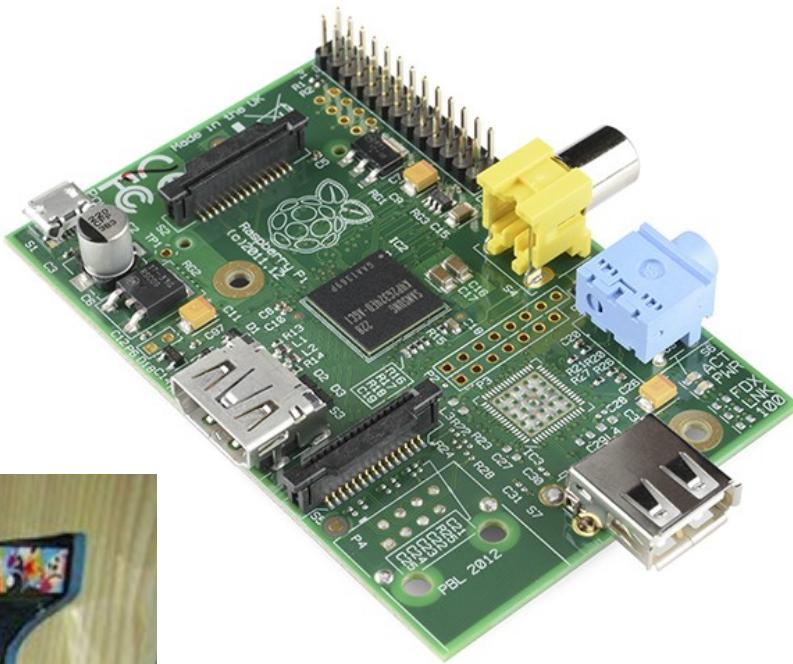


- **Microcontrolador** de 8 bits ATmega2560 16 MHz
- Pinos **digitais** de entrada/saída: 54
- Pinos **analógicos** de entrada: 16
- Memória Flash: 256 KB (8KB usados pelo *bootloader*)
- SRAM: 8KB
- EEPROM: 4KB

\$13.48

# Raspberry Pi

FIAP

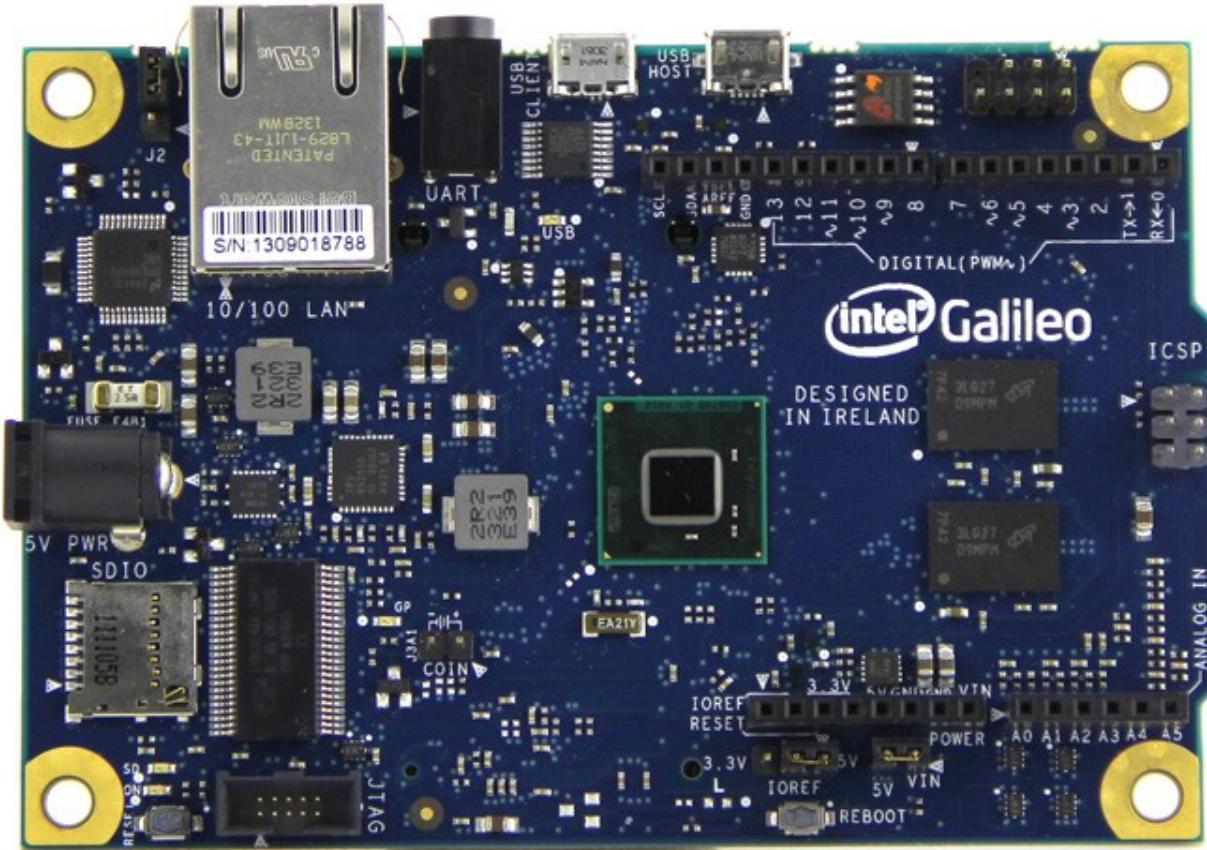


- **Processador** SoC – Broadcom BCM2836 *quad core* Cortex A7 @ 900MHz
- Memória do Sistema – 1GB SDRAM
- GPU VideoCore IV
- Saída de Vídeo e Áudio – HDMI e AV via 3.5mm jack
- Interface Ethernet 10/100M
- 4x portas USB 2.0 e 1x porta micro USB
- Expansão:
  - Conector para GPIOs com 2×20 pinos
  - Conector para Câmera
  - Conector para *Display*

\$44.95

# Intel Galileo e Edison

FIAP



# Intel Galileo e Edison – Especificações

FIAP

## Galileo:

Intel Quark System-on-Chip (SoC) x1000  
com um core de 32bit a 400MHz.

Conexões: mini-PCI Express (para o módulo Wifi-Bluetooth), porta 100Mb Ethernet, slot Micro-SD, porta serial RS-232, USB Host/Client.

Fator de forma: pins compatíveis com Arduino.

\$69.90

## Edison:

22 nm Intel SoC incluindo:

- Uma CPU dual-core, dual-threaded Intel Atom a 500 MHz
- Um microcontrolador 32-bit Intel Quark a 100 MHz

Memória: 1 GB LPDDR3

Armazenamento: 4 GB eMMC

Wifi: a/b/g/n

Bluetooth: 4.0 Low Energy

\$99.90

Arduino usa um **microcontrolador** de 8-bit. É simples e cobre as necessidades básicas de um programador.

Mas é impossível instalar um sistema operacional completo, como Linux, o que limita as suas possibilidades.

Para instalar um SO completo que tenha suporte a rede e a possibilidade de usar compiladores, faz-se necessário um **processador**.

Nos computadores são conectados dispositivos de alto nível de E/S, como USB. Mas a maioria dos sensores IoT são mais simples, e usam uma E/S analógica e digital de baixo nível.

Para que uma plataforma de **IoT** seja versátil, deve suportar **E/S analógica e digital**.

Algumas plataformas IoT só suportam E/S digital (como a Raspberry Pi), e outras como Arduino e Intel Galileo/Edison suportam os dois tipos (analógica e digital).

O suporte de rede é muito importante em IoT.  
É possível escolher uma plataforma sem suporte  
sem fio e adicionar um **dongle** para o protótipo,  
mas:

- É mais complexo passar para produção
  - As otimizações de consumo de energia são  
imitadas
  - A integração com software-SO não é imediata
- Dica: escolha plataformas com amplo suporte a rede  
para o protótipo e para produção.

# Consumo energético

Uma grande diferença entre protótipos IoT e soluções finais está no consumo de energia.

Ninguém ligaria um *dongle* 3G num computador e chamaria de smartphone né? Mesma coisa acontece com IoT.

É necessário:

- um processador altamente eficiente, com capacidades avançadas de hibernação
- consumo de energia sem fio otimizado
- boa integração entre todas as partes
- otimizações no software, nos drivers e no SO

O ideal seria fazer o protótipo com o nosso SO de escritório.

Pode ser uma distribuição Linux como Ubuntu, Windows 10, OSX... É fácil, e todos os pacotes estão disponíveis.

Mas um projeto de sistemas embutidos de grande envergadura requer começar do zero, tomindo uma série de medidas, possivelmente usando um substituto da biblioteca *glibc (GNU C Library)*, etc.

Um exemplo de SO de código aberto para projetos profissionais é **Yocto**.

# Arduino vs Raspberry

FIAP

	Arduino Uno	Raspberry Pi
	Microcontrolador	Microprocessador
CPU	AVR ATmega328p	Broadmcom ARM117
Velocidade	16 MHz	700 MHz
Registro	8 bit	32 bit
RAM	2 k	512 MB
GPU	-	VideoCore IV
Pinos	54+16	40
E/S	40 mA Digital e analógica	5-10 mA Digital
Consumo	175 mW	700 mW
SO	—	Linux (Raspbian)

# Arduino vs Raspberry

FIAP



**Arduino  
Microcontroller**

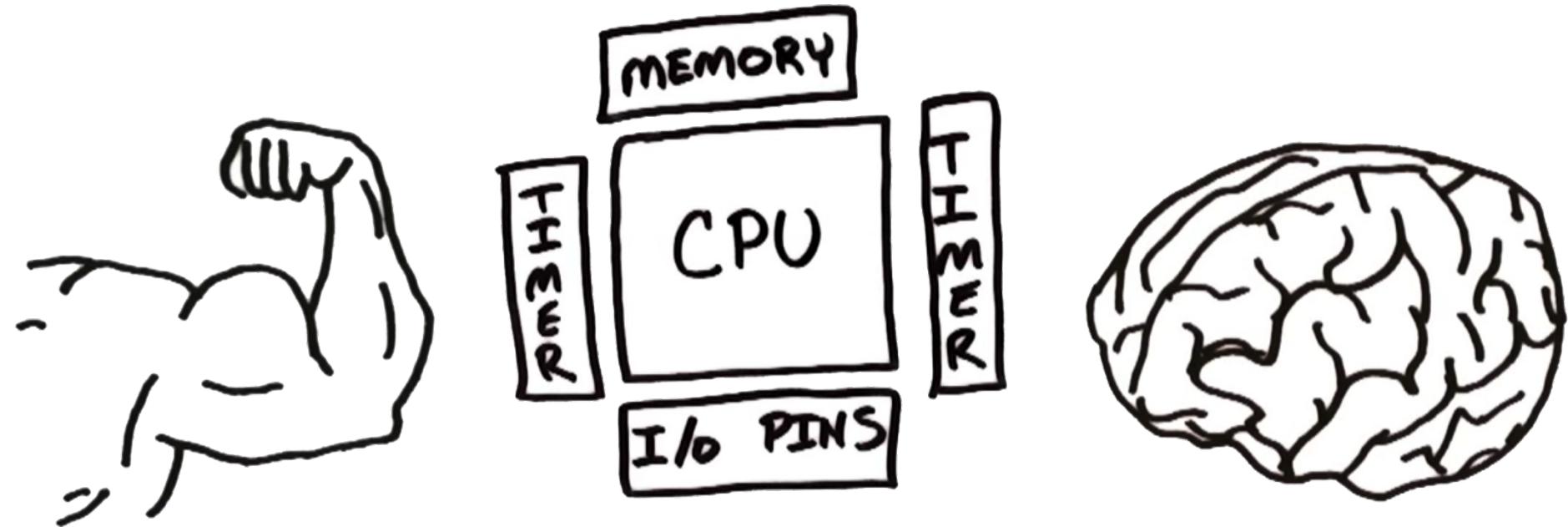


$$2+?=4$$

**Raspberry Pi  
Microprocessor**

# Arduino vs Raspberry

FIAP



**Microcontroller**

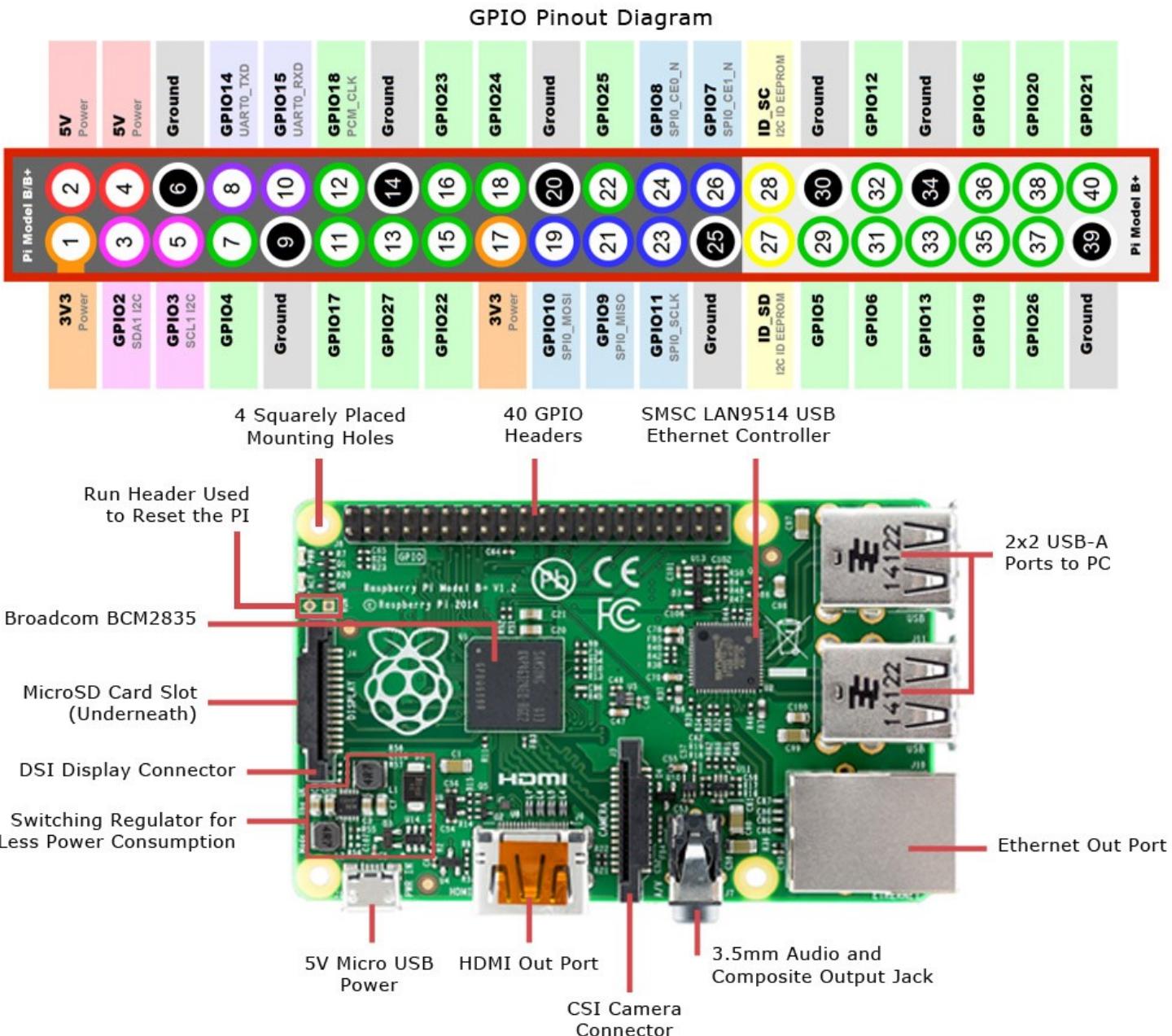
**Microprocessor**

*HandsOn Lab*  
**Configuração inicial da Raspberry Pi**

## 1. Primer contato com a Raspberry Pi – Configuração inicial da placa

# Raspberry Pi 2

FIAP



## 1. Primer contato com a Raspberry Pi

- Configuração inicial da placa

- Conexão a Internet

- WIFI:

- REDE WIFI EVENTOS

- 1234567890

## 2. Quadro virtual

*<https://notepad.pw/apicon>*

# **Plataformas de desenvolvimento – Métodos de programação**

- Compilação **cruzada**:
  - IDE Arduino
- Desenvolvimento com Linux **na própria placa**

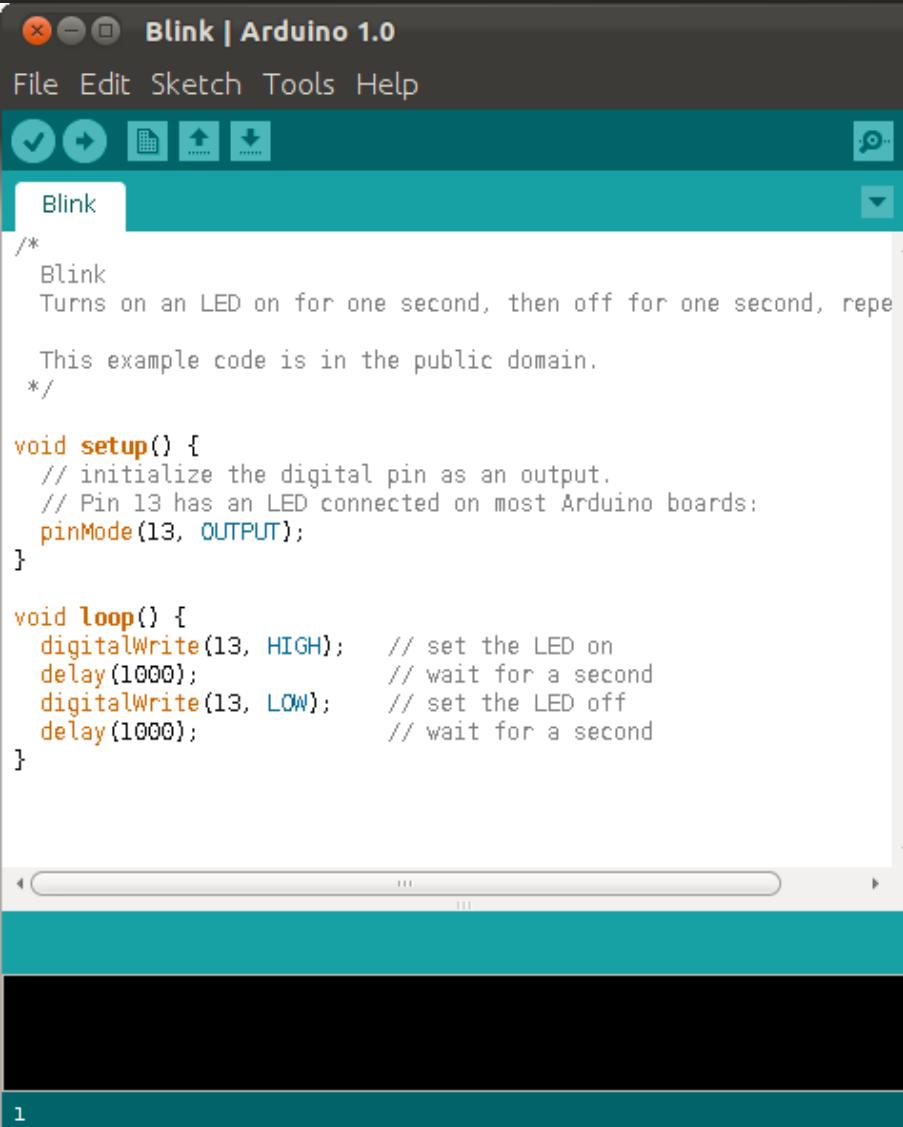
## *Cross-compiler*

Capaz de gerar código executável para uma plataforma diferente da qual o compilador está sendo executado.

Utilizado principalmente para gerar código para plataformas que não suportam ambiente de desenvolvimento como, por exemplo, **sistemas embarcados e microcontroladores**.

# IDE Arduino

FIAP



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.0". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for Save, Run, Upload, and Download. The main window displays the "Blink" sketch. The code is as follows:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repe
  This example code is in the public domain.
*/

void setup() {
  // initialize the digital pin as an output,
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);      // set the LED on
  delay(1000);                // wait for a second
  digitalWrite(13, LOW);       // set the LED off
  delay(1000);                // wait for a second
}
```

O desenvolvimento para **plataformas embarcadas** requer ferramentas que possibilitem a compilação do código e o **envio do binário** para a placa.

Arduino IDE disponibiliza de forma integrada:

- editor de código
- compilador cruzado
- loader*

Arduino possui uma série de **simuladores** que possibilitam o **desenvolvimento**/simulação de circuitos (**antes** de se aplicar às plataformas de desenvolvimento)

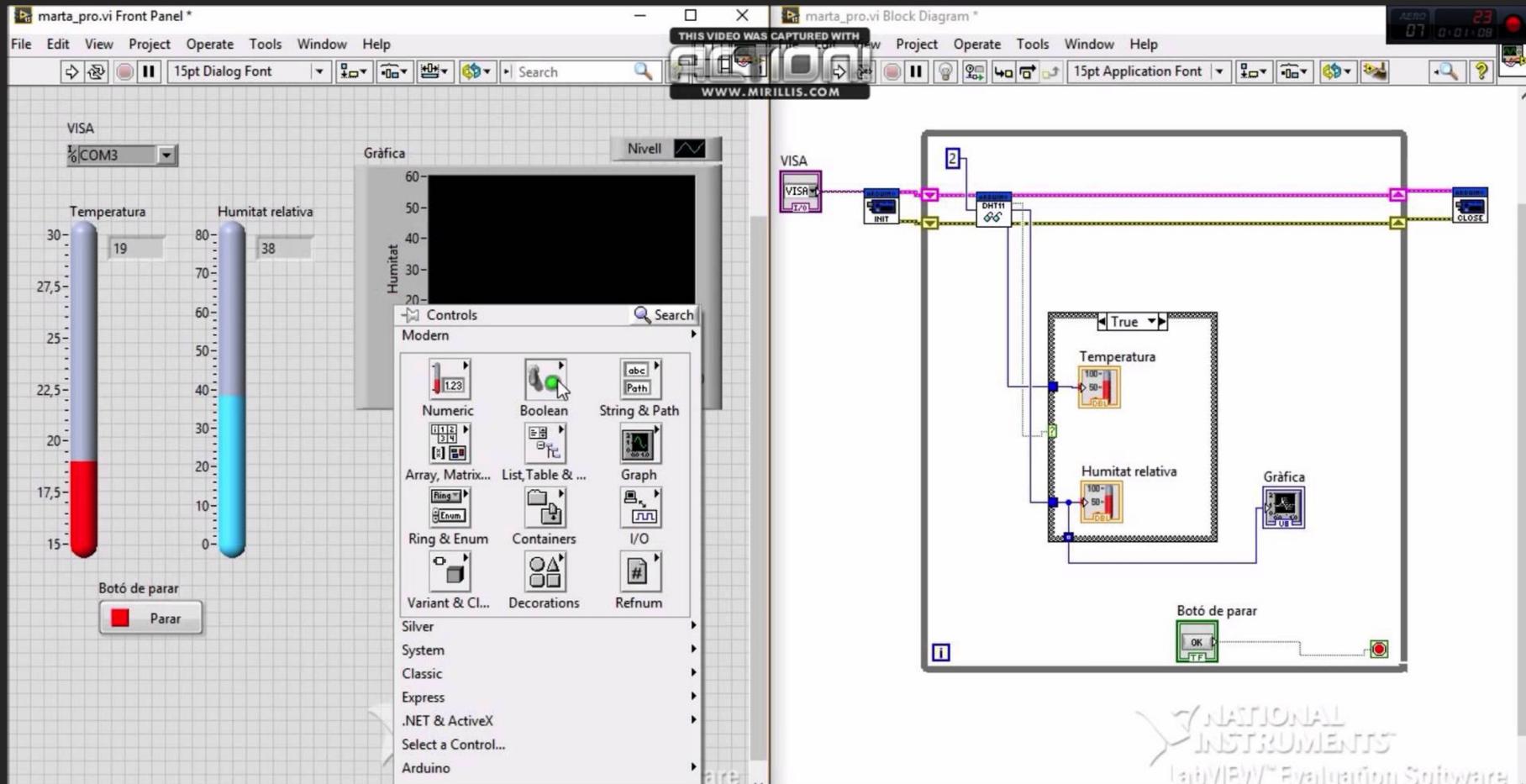
- elimina possibilidades de perda de equipamentos (por erros de desenvolvimento)
- possibilita o desenvolvimento por usuários que não possuem o dispositivo.

Alguns desses simuladores são:

- Arduino Toolkit
- Protheus

# IDE Arduino

FIAP



Acesso por *ssh* na placa desde a rede ou USB,  
instala os pacotes necessários,

*e programa com qualquer linguagem no Linux*

- Sem necessidade de instalar nada no PC
- Edição de código com *nano*, *emacs* ou *vi*
- Compilação com *gcc* na própria placa
- Possibilidade de usar VMs como *Python*, *NodeJS*, ...
- Possibilidade de interagir com serviços Linux
- Interação com SO (Yocto, Raspbian, ...)
- Possibilidade de construir uma distribuição a medida.

Público alvo :

- pessoas com experiência em Linux, ou em Raspberry Pi
- possibilidade de conectar qualquer sensor
- controle absoluto em baixo nível,  
necessário se for mudar para um ambiente de produção
- sem necessidade de instalar nada no PC

**Baixo nível, sem limites, estilo Linux**

Para migrar o código:

- **Arduino** : não é fácil migrar o código para outro entorno
- **Linux** : possível migração a um entorno de produção e a um desenvolvimento profissional de sistemas embutidos / IoT

# | Links

FIAP

Google Nest <https://nest.com/>

Sen.se mother <https://sen.se/mother/>

Mash Machine <http://mashmachines.com/>

Grove [http://www.seeedstudio.com/wiki/Grove\\_System](http://www.seeedstudio.com/wiki/Grove_System)

MBA<sup>+</sup>

Copyright © 2017 Prof. MSc. José Castillo Lema

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

Um computador consome o 90% da sua atenção:  
é uma ferramenta de uso exclusivo.

Um telefone moderno com tela de toque: 40%  
Pode caminhar, mas não dirigir de forma segura.

Um telefone antigo: 20%

Soluções IoT: visam 0-5% no máximo

## Por que um máximo de **0-5%**?

- Pode ter 50 dispositivos IoT em casa.  
Não vai ter tempo para dedicar a cada um deles
- Não precisa pensar ao interagir com os dispositivos IoT. Interfaces naturais são favorecidas
- A melhor interação: interação passiva

Sem tela, sem configuração

Um projeto não é considerado IoT, quando:

- não coleta informação do entorno mediante sensores analógicos ou digitais
- não produz ações no mundo real (mediante atuadores)

A ideia é interagir com o mundo real

Pode ser a Internet normal, mas também pode ser:

- Internet intermitente (para poupar energia, ou porque não está disponível)
- Baixa largura de banda, distâncias grandes – Sigfox, LoRaWAN
- Redes *Mesh*
- Rede local : Bluetooth LE, ZigBee

A “Internet” para IoT não é a Internet normal

I Inteligente não significa que a informação seja inteligente por si mesma.

Uma solução IoT se diz inteligente quando:

- Faz um análise cruzada dos dados sensoriais
- É realizado um primeiro nível de IA localmente
- A solução possui uma IA central
- Funciona em cenários com conectividade intermitente

Não espere pelo sensor perfeito

Esperam-se projetos com milhares/milhões de dispositivos conectados

Dispositivos IoT > Telefones móveis em 2018  
20 bilhões dispositivos em 2020

Novas técnicas para:

- Análise de grandes quantidades de dados – *big data*
- Novos paradigmas de rede