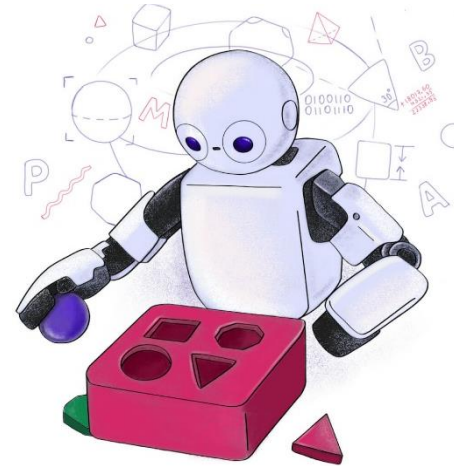


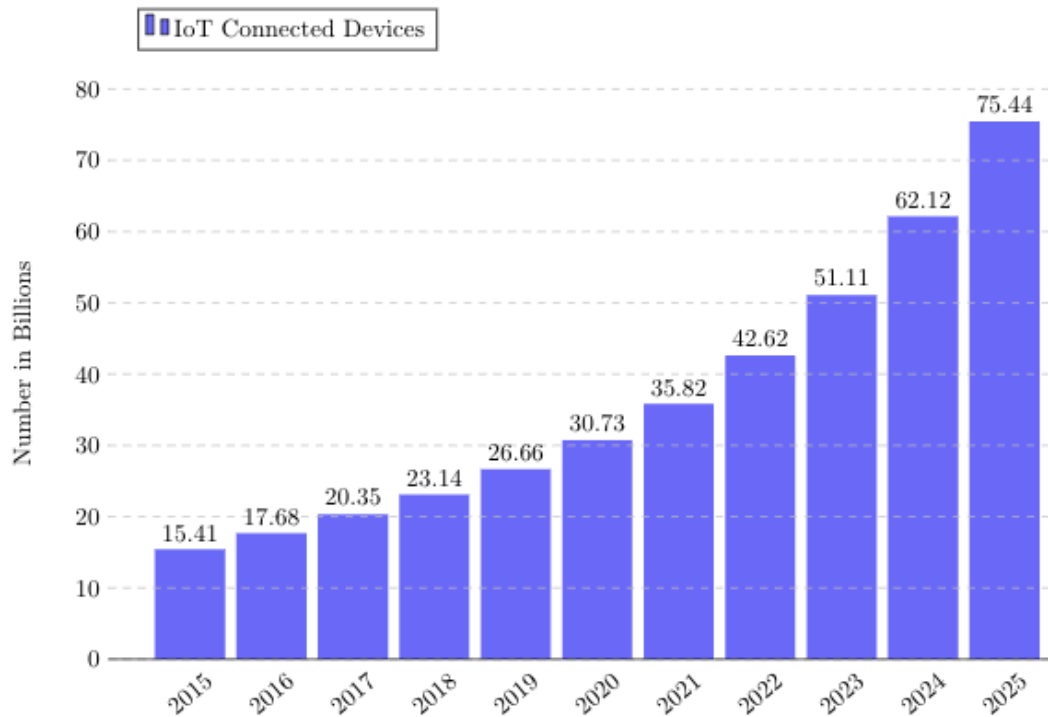
# TP557 - Tópicos avançados em IoT e Machine Learning: *Introdução ao curso*



***Inatel***

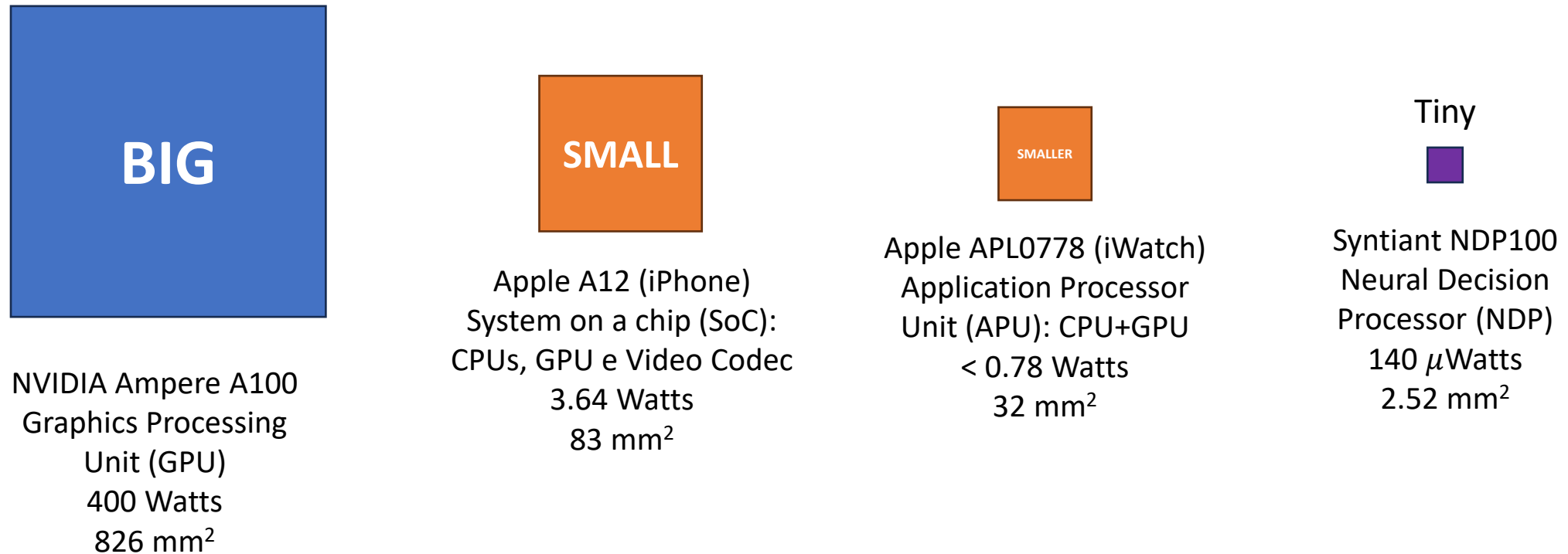
Felipe Augusto Pereira de Figueiredo  
felipe.figueiredo@inatel.br

# Motivação



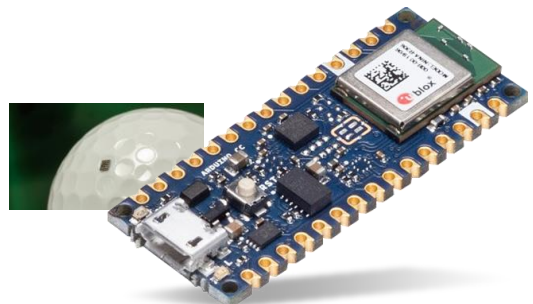
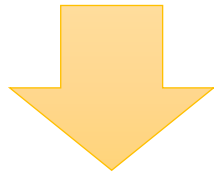
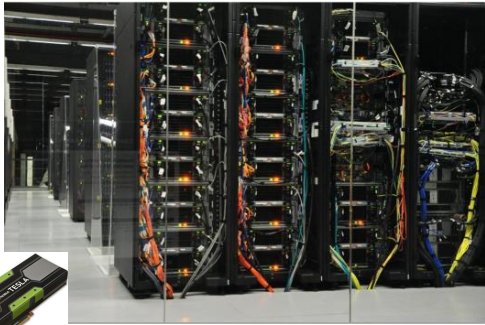
- Estima-se que teremos mais de 75 bilhões de dispositivos IoT até 2025.
- A grande maioria desses dispositivos estão equipados com microprocessadores ou microcontroladores e (vários) sensores.

# Motivação



- Se comparados com servidores, equipados com dezenas de CPUs e GPUs, os processadores e microcontroladores desses dispositivos são muito menores, consomem muito menos energia e são mais baratos, devido a alta demanda.

# Motivação



- O baixíssimo consumo de energia possibilita que os dispositivos:
  - Funcionem por longos períodos de tempo sem trocar a bateria (e.g., anos);
  - E executem aplicações “*Always On*”:
    - ✓ Aplicação constantemente ativa e monitorando o ambiente ao seu redor, coletando informações, as processando e respondendo em tempo real, sem interrupção.
    - ✓ Característica muito importante em dispositivos que exigem monitoramento constante, como sistemas de segurança, monitoramento de saúde, sistemas de automação industrial ou dispositivos de assistência virtual.

# Motivação

- Dispositivos IoT, também chamados de ***endpoints***, possuem muitas vezes uma grande variedade de sensores!

## Sensores de movimento

Giroscópios, radares (e.g., LiDAR), magnetômetros, acelerômetros.

## Sensores de acústicos

Microfones, ultrassônicos, vibrômetros, geofones.

## Sensores ambientais

Temperatura, humidade, pressão, infra-vermelho.

## Sensores de toque

Capacitivos, Resistivos.

## Sensores de imagem

Térmicos, RGB, LiDAR.

## Sensores biométricos

impressão digital, batimento cardíaco, glicose.

## Sensores de força

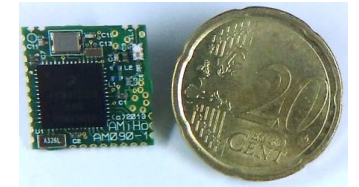
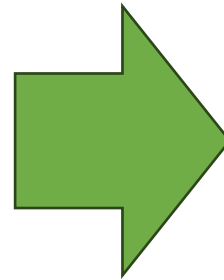
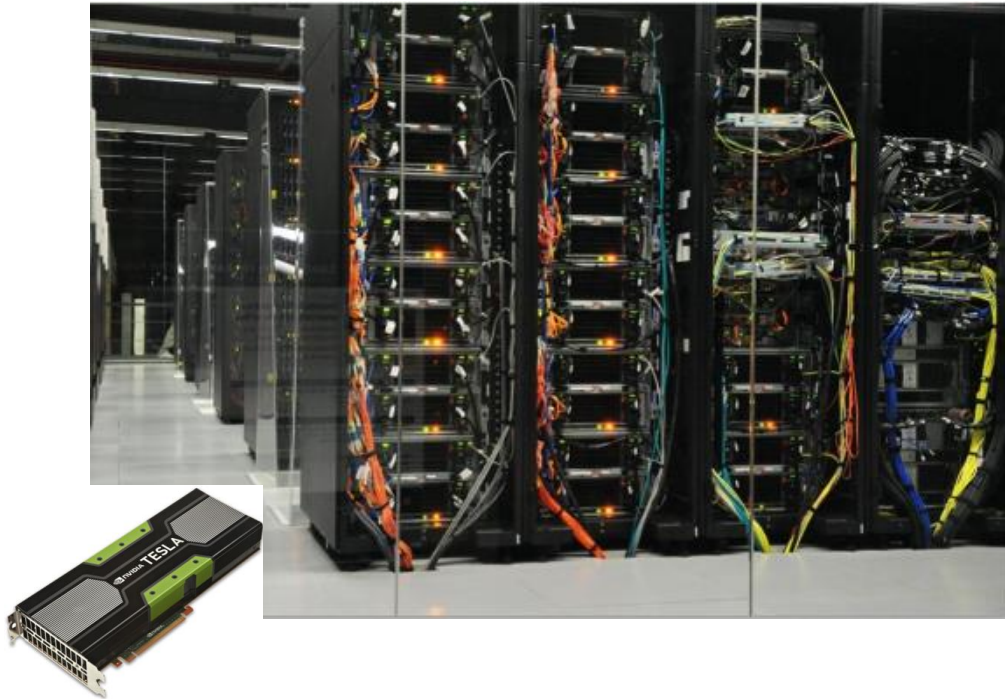
Pressão, deformação/tensão.

## Sensores de rotação

Encoders, potenciômetros.



# Motivação



***E se nós pudéssemos executar aplicações Always-On baseadas em inteligência artificial (IA), mais especificamente machine learning (ML), nesses dispositivos?***

***O que poderíamos fazer?***

# EdgeML versus TinyML

## Hardware



### TinyML

- CPU de baixo consume (32 bits single core)
- Pouca memória (RAM/Flash)
- Bateria com baixa capacidade
- Foco na longa duração da bateria

Anomaly Detection  
Sensor Classification  
20 KB

KeyWord Spotting  
Audio Classification  
50 KB

Image  
Classification  
250 KB+



Rpi-Pico  
(Cortex-M0+)



Arduino Nano  
(Cortex-M4)



Arduino Pro  
(Cortex-M7)



## EdgeML

### EdgeML

- CPU mais poderosa (64 bits e + cores)
- GPU
- + Memória (RAM/Flash)
- Bateria com maior capacidade
- Sem foco na longa duração da bateria

Object Detection  
Complex Voice  
Processing  
1 MB+

Video  
Classification  
2 MB+



RaspberryPi  
(Cortex-A)



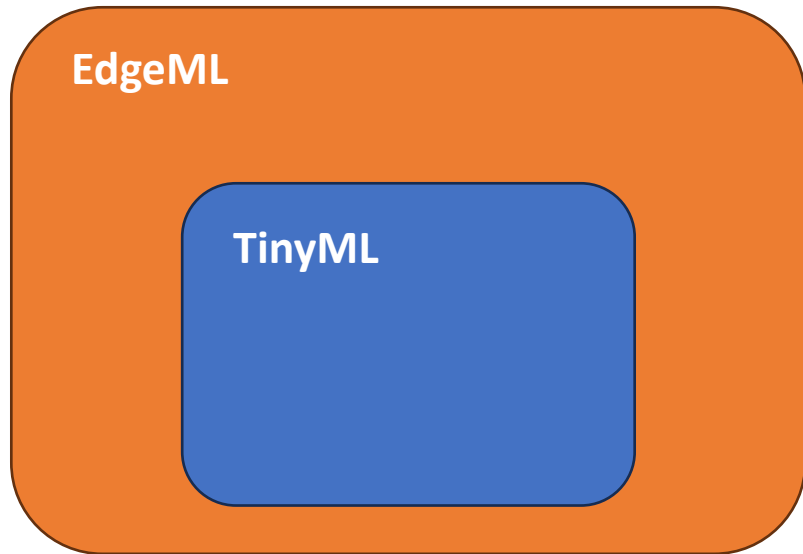
SmartPhone  
(Cortex-A)



Jetson Nano  
(Cortex-A + GPU)

Diferenciação  
de hardware e  
terminologia

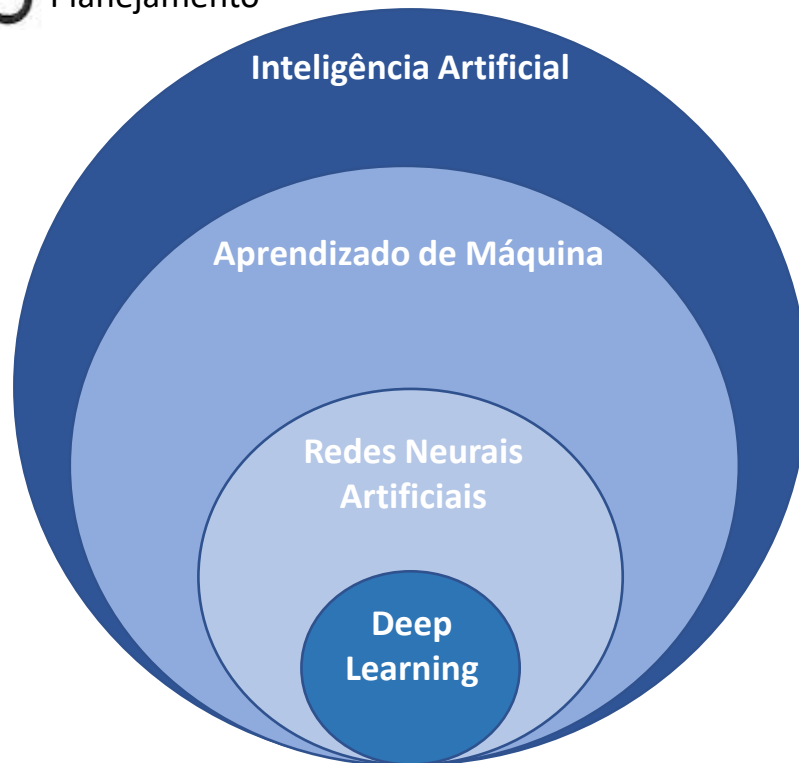
# EdgeML versus TinyML



- **Edge Computing:** processamento e armazenamento de dados é feito próximo ou nos dispositivos e sensores que coletam os dados (ou seja, na borda) e não na nuvem.
- **EdgeML:** processamento de algoritmos de IA na borda, ou seja, nos dispositivos e sensores.
- **TinyML:** subconjunto do **EdgeML**, onde os dispositivos e sensores coletam e processam dados com consumo de energia ultrabaixo (baterias), possibilitando a execução contínua de modelos de ML (i.e., dispositivos “**always on**”).



# Mas o que são IA e ML?

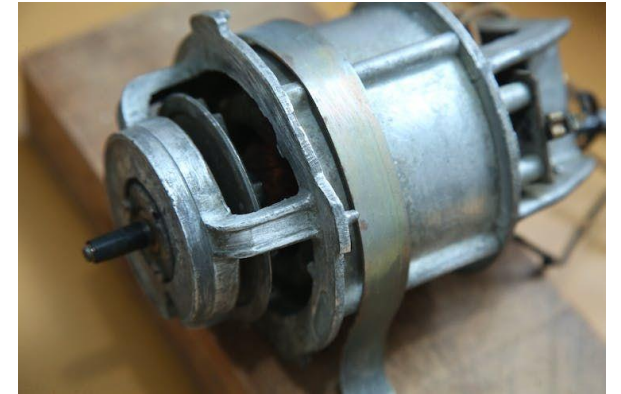


- **IA:** técnicas que façam as máquinas imitarem o comportamento humano.
- **ML:** algoritmos que aprendem uma tarefa sem serem explicitamente programados para tal.
- **RNAs:** modelos inspirados no funcionamento dos neurônios do cérebro humano. Consistem em neurônios artificiais interconectados, que são organizados em camadas e capazes de aprender a partir de dados.
- **DL:** modelos que extraem (i.e., aprendem) padrões complexos de dados usando redes neurais com várias camadas (i.e., profundas).
  - Por terem maior capacidade, necessitam de grandes quantidades de dados para aprender.

# Aplicações

## Manutenção preventiva

- **Aplicação:** monitoramento contínuo de equipamentos em busca de mau funcionamento e prever falhas antes que elas aconteçam.
- **Sensores que podem ser utilizados:** movimento, corrente, áudio, câmera.



# Aplicações

## Monitoramento e rastreio de bens

- **Aplicações:**
  - Monitoramento da saúde de animais e lavouras.
  - Classificação de doenças do café.
  - Detecção de incêndios.
- **Sensores que podem ser utilizados:**  
movimento, temperatura, humidade, posição, áudio, câmera.



# Aplicações

## E-health

- **Aplicações:**
  - Detecção de fibrilação arterial em sinais eletrocardiograma.
  - Detecção de doenças pulmonares (e.g., COVID, pneumonia).
  - Detecção de apneia do sono.
  - Detecção de posturas incorretas.
  - Previsão de Hospitalização por Insuficiência Cardíaca
- **Sensores que podem ser utilizados:** movimento, áudio, imagem e ECG.





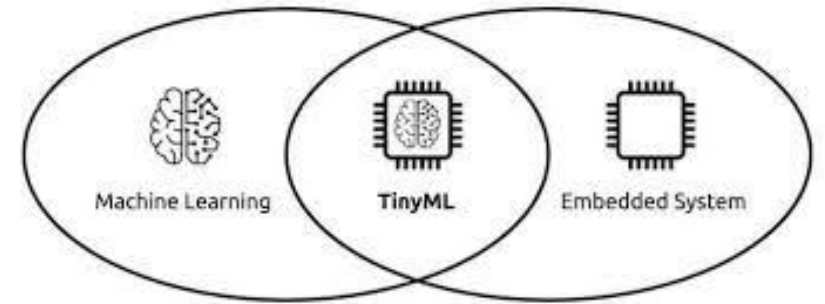
# Sobre o curso

- A difusão de dispositivos de **(ultra)baixo consumo de energia**, equipados com sensores, juntamente com a introdução de **bibliotecas de IA para microcontroladores**, têm permitido a proliferação em massa de dispositivos de **Internet das Coisas (IoT) Inteligentes**, ou seja, que executam algoritmos de ML.
- Esse novo paradigma é chamado de **TinyML: IoT + Sensores + IA (ML)**.



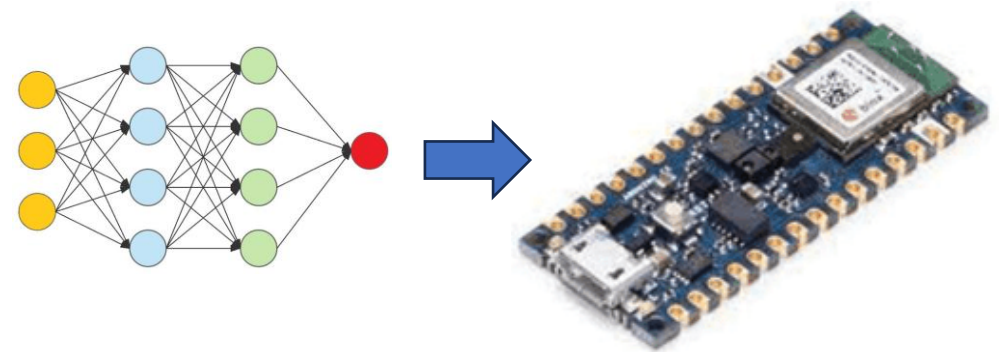


# Sobre o curso



- **TinyML**: subárea da IA que lida com a *implementação de algoritmos, técnicas e ferramentas otimizadas para a execução de modelos de ML em dispositivos de baixo consumo de energia*, como sensores IoT, wearables e dispositivos portáteis.
  - Envolve a *minimização do tamanho do modelo*, o *uso de algoritmos de ML leves* e a *otimização dos recursos computacionais* dos dispositivos.
- O processamento local dos dados em dispositivos de borda traz benefícios como
  - **Menor latência**: não é necessário o envio dos dados para um servidor na nuvem, possibilitando aplicações que operem em tempo real;
  - **Maior privacidade**: dados sensíveis não são transferidos para servidores;
  - **Menor dependência de conexões**: reduz ou elimina a necessidade de transferência de dados para servidores.
  - **Maior autonomia energética**: o uso de microcontroladores com baixo consumo de energia reduz o consumo de energia, uso de recursos e custos.
  - **Reduzido consumo de memória**: modelo e binário (i.e., executável) menores.

# Sobre o curso



- Portanto, o curso mistura ML com dispositivos IoT, cujas principais características são:
  - **Baixo poder computacional:** geralmente equipados com CPUs de 32 bits, poucos quilobytes de memória RAM e alguns megabytes de memória de longo prazo.
  - **Baixíssimo consumo de energia:** são geralmente alimentados à bateria, as quais devem durar um longo período de tempo.
  - **Sensores:** temperatura, pressão, humidade, ópticos (e.g., câmeras RGB, térmicas), movimento (e.g., giroscópio e acelerômetro), biométricos (e.g., impressão digital, batimento cardíaco), etc.
- Nosso curso é baseado nos cursos de ***TinyML*** de Harvard e da UNIFEI.
  - <https://scholar.harvard.edu/vijay-janapa-reddi/classes/cs249r-tinyml>
  - <https://github.com/Mjrovai/UNIFEI-UESTI01-TinyML-2022.1>

# Sobre o curso



TensorFlow Lite



- O curso será dividido em duas partes:
  - **Fundamentos:** Introdução e desafios do TinyML, paradigmas de ML, introdução ao *Deep Learning*, modelos de DL para regressão e classificação, redes convolucionais, como evitar problemas de sobreajuste e introdução ao **Edge Impulse**;
  - **Aplicação prática dos fundamentos:** ciclo de vida e de trabalho de aplicações de ML, introdução às bibliotecas **TFLite** e **TFLite-Micro**, visão geral do kit de desenvolvimento, laboratórios práticos usando um kit de desenvolvimento (ou celulares pessoais) e modelos de ML e apresentação dos trabalhos.
- **Pré-requisitos** (*não mandatórios*):
  - Conceitos básicos de **álgebra linear** (e.g., matrizes e vetores), **cálculo** (e.g., algoritmos de otimização, como o gradiente descendente), **probabilidade e estatística** (e.g., distribuições de probabilidade, média, desvio padrão, validação cruzada) e **processamento de sinais** (e.g., FFT, filtragem);
  - Conhecimentos básicos de programação em C/C++ e Python.

# Sobre o curso

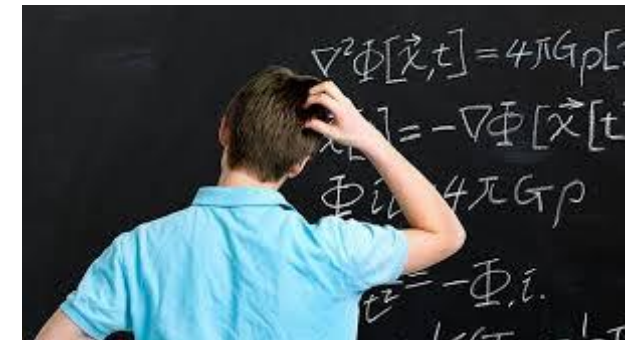


## Avaliações

- Atividades (A): 30%
  - Exercícios de programação
  - Quizzes
  - Proposta do projeto final (ao final do segundo mês)
- Relatórios dos Laboratórios (R): 30%
  - Cinco (5) laboratórios com experimentos práticos.
- Projeto final (P): 40%
  - Projeto prático unindo IoT e ML.
  - Até dois (2) alunos por projeto.
  - Tema será escolhido por vocês.

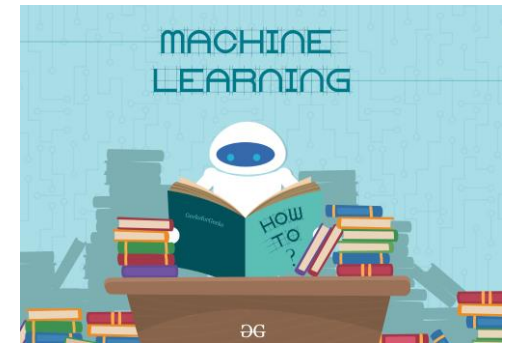


$$NF = A * 30\% + R * 30\% + P * 40\%$$



# Objetivos do curso

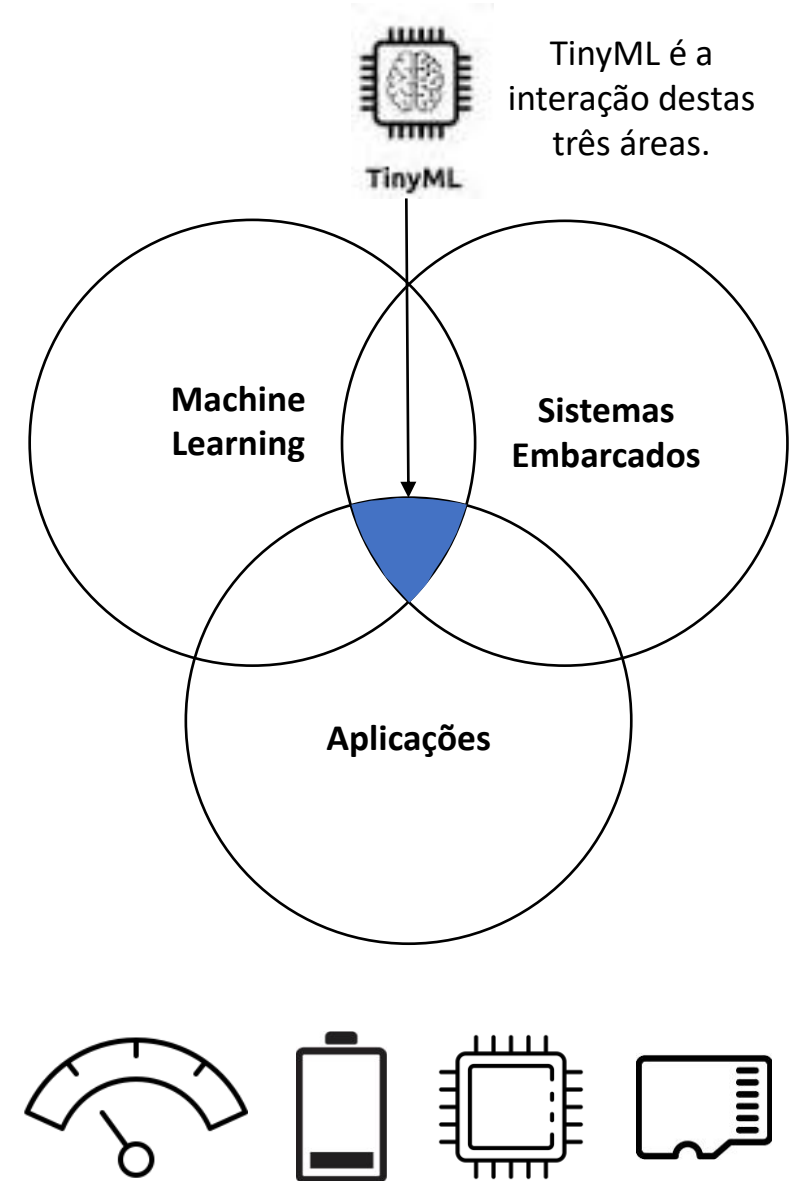
- Apresentar os fundamentos de ML e IoT, bem como explorar a interseção entre essas áreas.
- Familiarizar os alunos com a literatura e os conceitos-chave da área de TinyML.
- Capacitar os alunos a treinar e implementar modelos de ML em smartphones e dispositivos IoT.
- Desenvolver a capacidade dos alunos de conceber e criar aplicações TinyML que funcionem em smartphones e dispositivos IoT.
- Proporcionar uma experiência prática por meio de laboratórios e projetos práticos, permitindo que os alunos apliquem seus conhecimentos de TinyML em cenários do mundo real.





# Quais áreas vamos aprender?

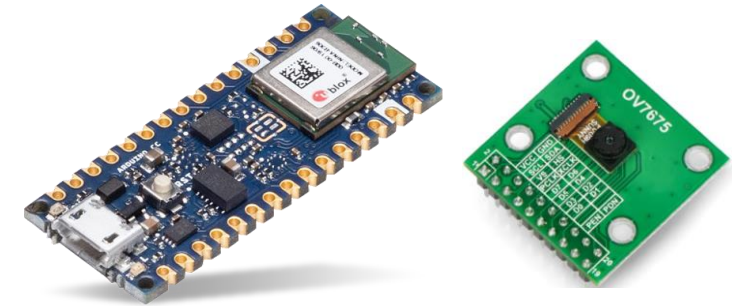
- Aprenderemos os ***fundamentos*** de cada uma dessas áreas, mas apenas o ***suficiente*** para focarmos na construção de ***aplicações de TinyML***.
- Trabalharemos na interseção dessas três áreas, levando em consideração algumas questões importantes:
  - Como o ML pode viabilizar novas e interessantes aplicações práticas de TinyML?
  - Quais são os desafios de implementar ML em dispositivos embarcados minúsculos e com recursos limitados?
    - ✓ Temos que considerar problemas de latência (e.g., tempo para classificação de objetos em uma imagem) e restrições de bateria, processamento e memória.
  - Quais novos casos de uso podem ser habilitados com o uso de TinyML, que não eram possíveis antes?



# O que vamos usar?

## Hardware

- Smartphones
- Arduino Nano 33 BLE Sense
- Sensores



## Software

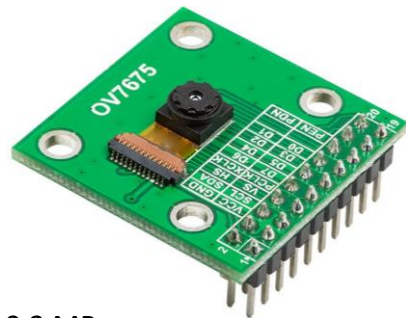
- **Biblioteca de ML:** Tensorflow (TFLite e TFLite Micro)
- **Ambientes de programação:** Google Colab (com Jupyter notebooks) e Arduino IDE.
- **Ambiente de desenvolvimento de ML:** Edge Impulse Studio.



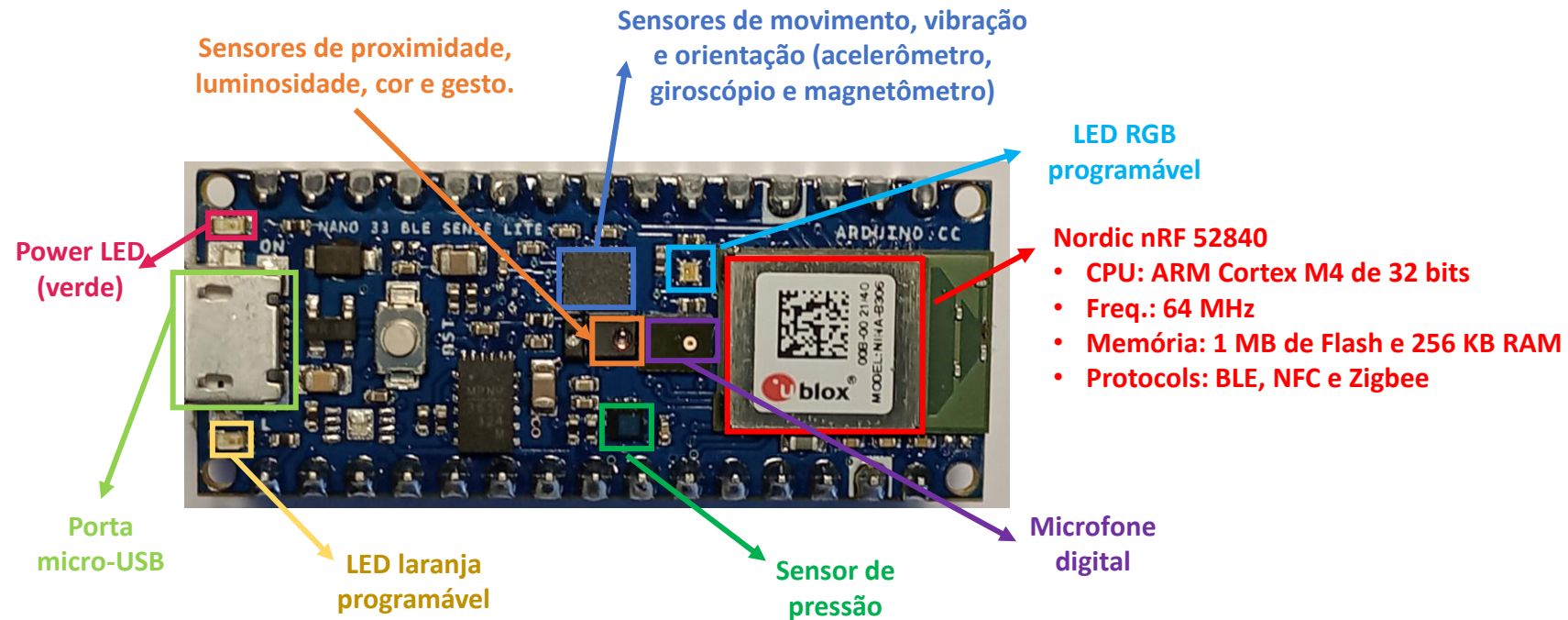
# O kit TinyML

## O kit contém:

- 1 x Arduino Nano 33 BLE Sense Lite
- 1 x Shield
- 1 x Câmera OV7675
- 1 x Cabo USB A



Câmera de 0.3 MP  
**Resoluções (pixels):** 640x480 (VGA), 320x240 (QVGA) e 160x120 (QQVGA)  
**Frames por segundo:** 30 (VGA), 60 (QVGA) e 240 (QQVGA)

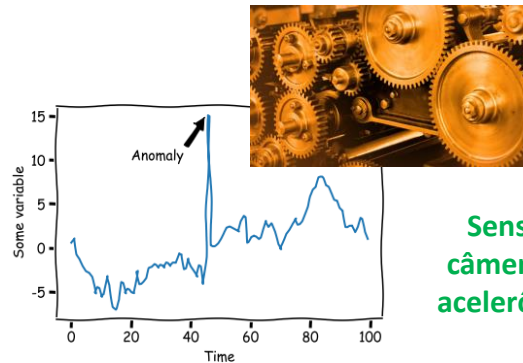


# Atividades práticas



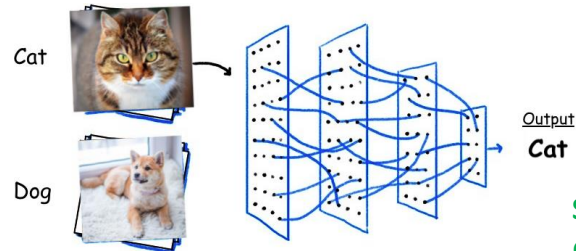
Classificação de movimento ou atividade

Sensores: acelerômetro, giroscópio, magnetômetro



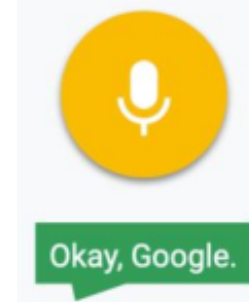
Detecção/previsão de anomalias

Sensores: câmera e/ou acelerômetro.



Classificação de imagens

Sensor: câmera



Keyword spotting

Sensor: microfone.



Visual wake words

Sensor: câmera.



Classificação de gestos

Sensores: câmera e/ou acelerômetro.



# Referências

- [1] Daniel Situnayake and Pete Warden, *"TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers"*, 1st ed., O'Reilly Media, 2019.
- [2] Aurélien Géron, *"Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems"*, 1st ed., O'Reilly Media, 2017.
- [3] Agus Kurniawan, *"IoT Projects with Arduino Nano 33 BLE Sense: Step-By-Step Projects for Beginners"*, Apress, 2020.
- [4] Stuart Russell and Peter Norvig, *"Artificial Intelligence: A Modern Approach"*, Prentice Hall Series in Artificial Intelligence, 3rd ed., 2015.
- [5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *"Deep Learning"*, MIT Press, 2016.
- [6] Hakima Chaouchi, *"The Internet of Things: Connecting Objects"*, Wiley, 2010.
- [7] Jun Zheng and Abbas Jamalipour, *"Wireless Sensor Networks: A Networking Perspective"*, Wiley, 2009.
- [8] [Coleção de livros](#)



# Avisos

- Todo material do curso está disponível no GitHub:
  - <https://github.com/zz4fap/tp557-iot-ml>
- Google Colab + Python Crash Course
  - [https://www.youtube.com/watch?v=inN8seMm7UI&ab\\_channel=TensorFlow](https://www.youtube.com/watch?v=inN8seMm7UI&ab_channel=TensorFlow)
  - <https://www.youtube.com/watch?v=inN8seMm7UI>
  - <https://www.youtube.com/watch?v=pq4NNIYar9o&list=PLRc6ZYt68prVXAhwY1JD6DFc3BJGmJriq&pp=gAQBiAQB>
- Introdução ao TensorFlow
  - [https://www.youtube.com/watch?v=yjprpOoH5c8&ab\\_channel=TensorFlow](https://www.youtube.com/watch?v=yjprpOoH5c8&ab_channel=TensorFlow)
- Horário de Atendimento
  - Todas as quartas-feiras das 17:30 às 18:30.
  - Presencialmente ou remotamente.

# Ideias para projetos finais

1. [Identificação de falhas em rolamentos](#)
2. [Identificação de doenças respiratórias](#)
3. [Detectando falhas em automóveis através de seu som](#)
4. [Identificação de pragas](#)
5. [Identificação de posturas incorretas](#)
6. [Inspeção automática de qualidade](#)
7. [Classificador de Vogais em Libras](#)
8. [Detecção de doença pulmonar](#)
9. [Detecção de Algarismos em Hidrômetros](#)
10. [Detecção de Roncos](#)
11. [Detecção de tumores cancerígenos](#)
12. [Detecção e prevenção de abalos sísmicos](#)
13. [Detecção de incêndios](#)
14. [Detecção de COVID através da tosse](#)
15. [Detecção do uso de máscaras](#)
16. [Personal Trainer](#)
17. <https://www.tensorflow.org/lite/examples?hl=pt-br>
18. <https://experiments.withgoogle.com/collection/tfliteformicrocontrollers>

# Atividades

- Quiz: “***TP557 – Introdução***”
- [Exercícios sobre programação em Python.](#)

Perguntas?

Obrigado!