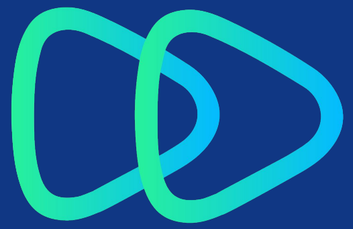


Portando Modelos IA para Microcontroladores de 32 bits

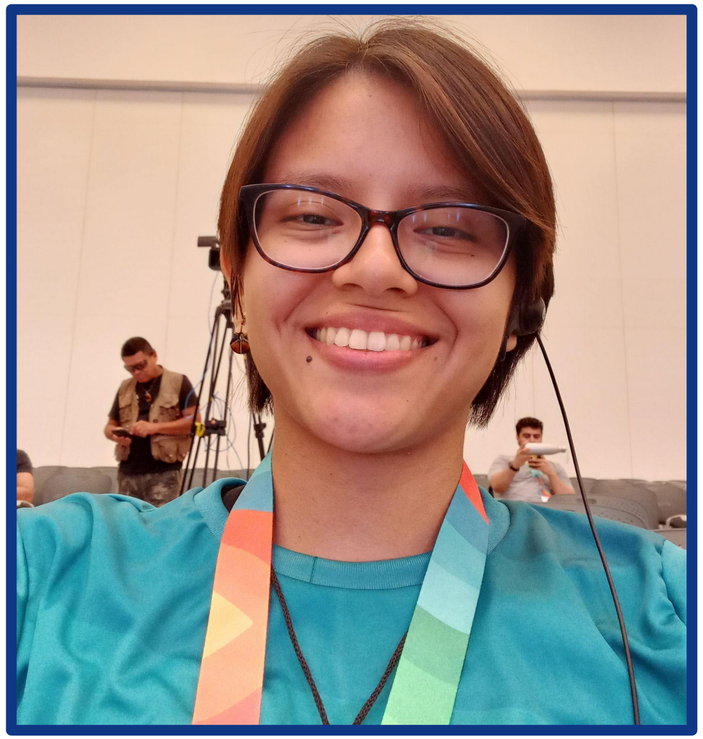
Lahis Almeida

15/10/2023

INDT - Instituto de Desenvolvimento Tecnológico

INDT

Instrutora



Lahis Almeida

Formação

- Bsc. Engenharia de Computação (**UEA**)
- Msc. Ciência da Computação (**Unicamp**);
- Desenv. de Sistemas Embarcados no **INDT.**

Experiência

- Internet das Coisas;
- Sistemas Embarcados;
- Edge IA.

Sumário

1 INSTITUCIONAL

2 CONCEITOS INICIAIS

3 KIT DE DESENVOLVIMENTO

4 EXP1 : HELLO WORD!

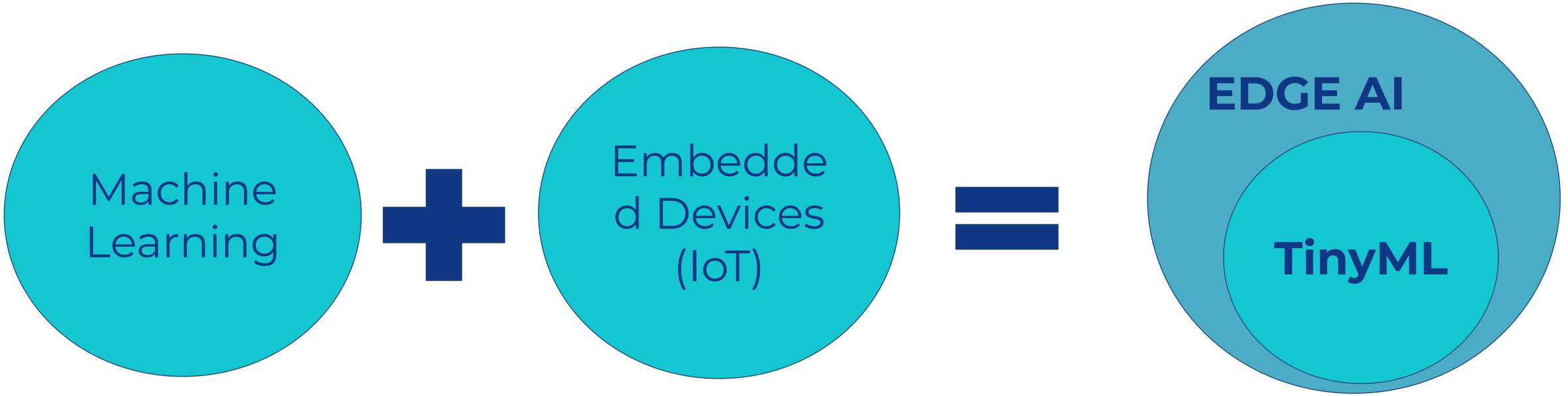
5 EXP2: CLASSIFICAÇÃO DE IMAGENS

6 PRÓXIMOS PASSOS

Institucional

- **O INDT está há mais de 20 anos oferecendo soluções em P&D e inovação no Brasil;**
- **Áreas de atuação:**
  - Indústria 4.0;
  - Laboratórios;
  - Telecom;
  - Saúde;
  - Mobilidade.
- **Formatos de parceria:**
  - Lei da Informática, EMBRAPII e FINEP
  - Contração direta.

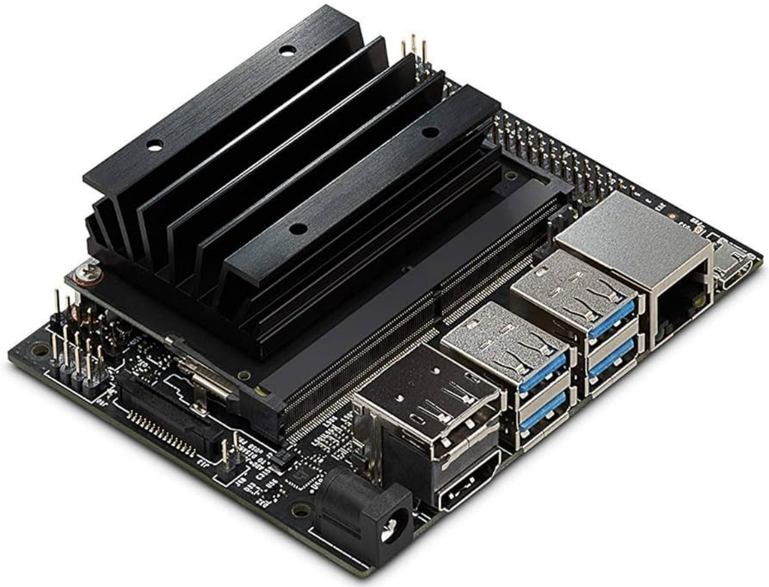
- **O que é TinyML?**
  - TinyML se refere a prática de implantar modelos de ML em dispositivos com restrição de recursos (MCUs);



- **O que é TinyML?**
  - Dispositivos IoT capazes de executar tasks IA localmente sem depender de um serviço de nuvem:
    - MCUs;
    - Wearables;
    - Sensores inteligentes

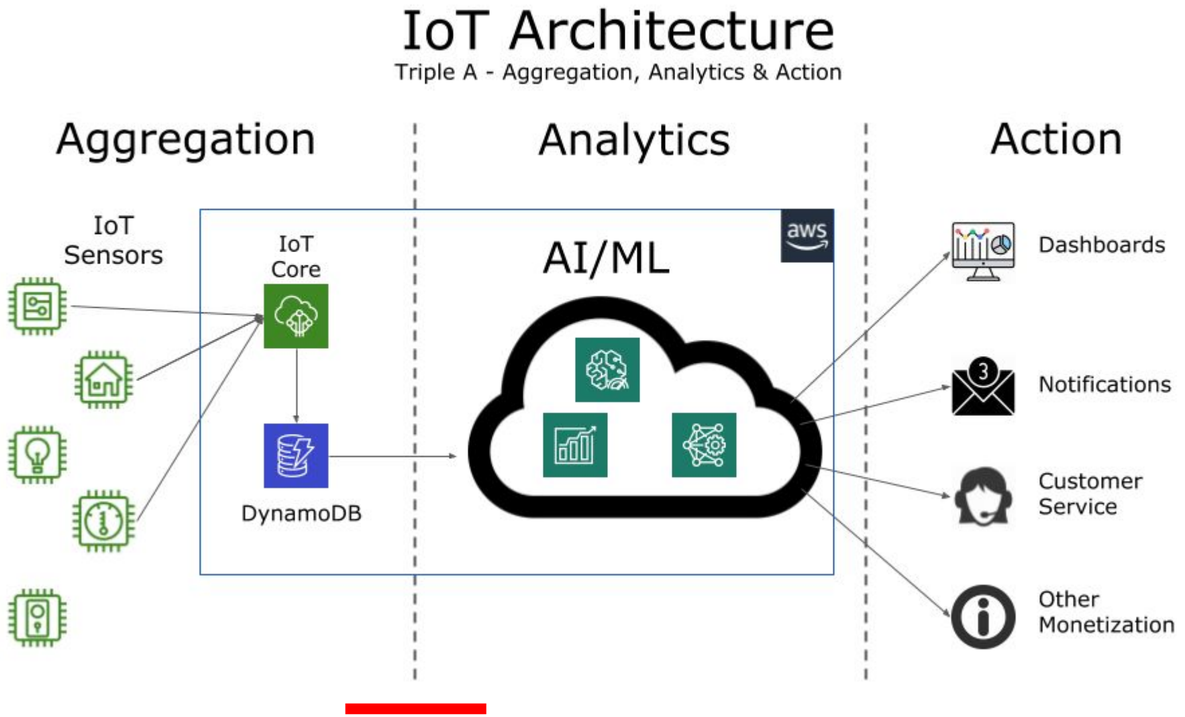
Seeeduino XIAO

ESP32



NVIDIA
Jetson NAno

- **Vantagens do TinyML**
  - Baixa Latência;
  - Segurança e Privacidade;
  - Eficiência em:
    - Largura de Banda
    - Consumo de Energia
  - Custo benefício.



- **Desafios do TinyML**
  - Restrição de recursos;
  - Complexidade dos Modelos;
  - Compatibilidade entre HW e SW.

- **Aplicações:** Domótica, Saúde, Indústria 4.0, Agricultura

- **Ecossistema de TinyML**
  - Arduino IDE;
  - **MicroML generator**;
  - EDGE Impulse;
  - TensorFlow Lite para Microcontroladores (TFLite Micro).
  - Compatibilidade entre HW e SW.

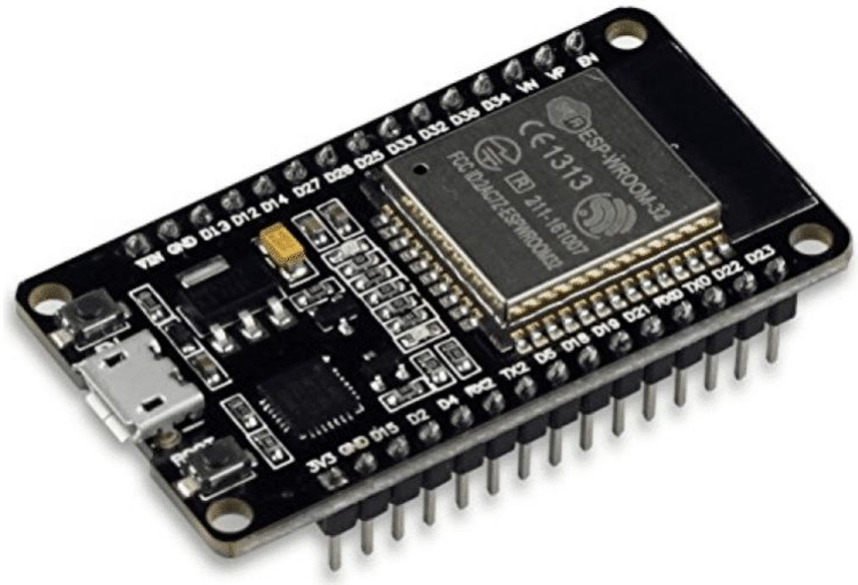
micromlgen

TensorFlow Lite

LABORATÓRIO

KIT DE DESENVOLVIMENTO

- **Kit de Desenvolvimento: ESP32**



Kit de Desenvolvimento ESP-WROOM-32

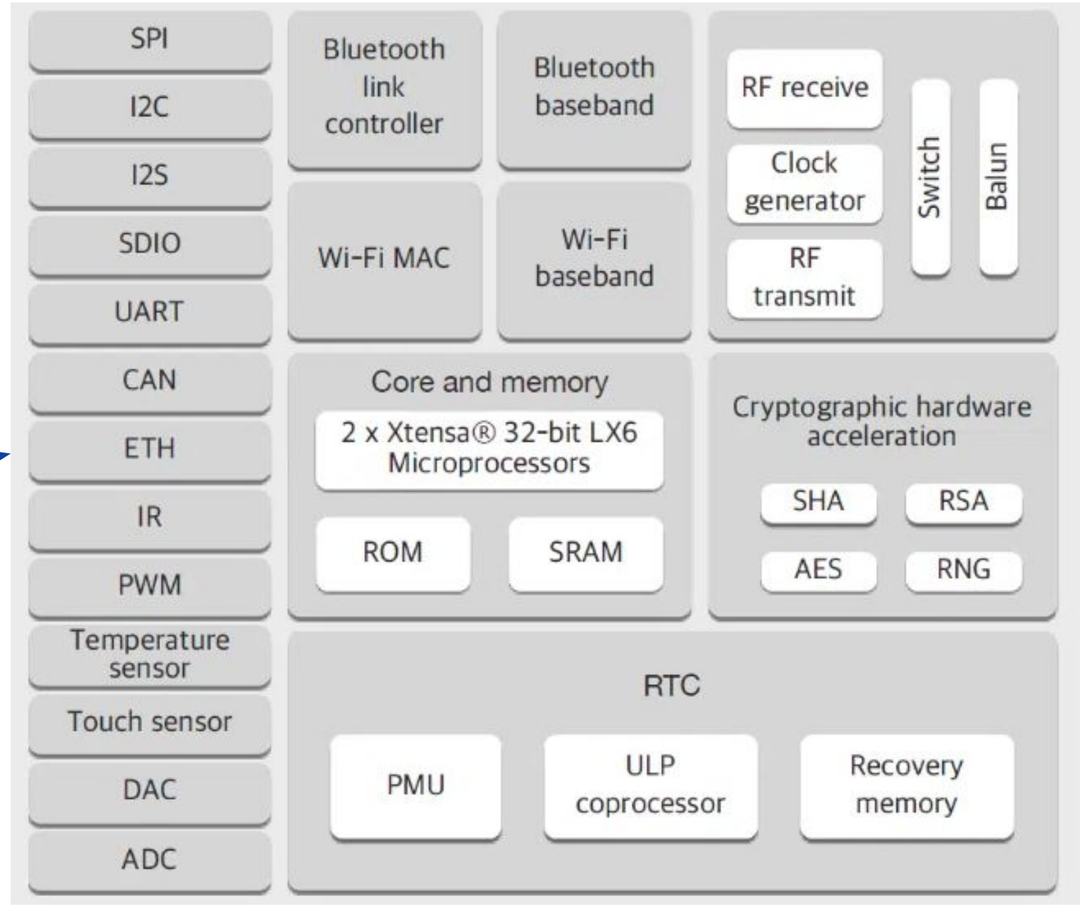


Diagrama de Blocos do chip ESP32

EXP I: Hello World!!

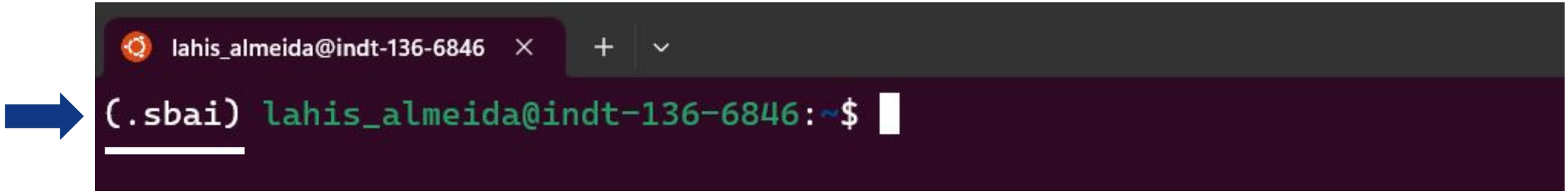
- **Objetivo:**
  - Obter familiaridade com ferramentas do minicurso
    - Sistema Operacional: **Ubuntu 22.04 LTS**
    - Ambiente Virtual Python: **.sbai**
    - Kit de desenvolvimento **ESP32**
    - **Arduino IDE**
  -

EXP I: Hello World!!

- **1. Acesso ao ambiente virtual .sbai:**
  - Abram o terminal do Linux e executem os seguintes comandos:

```
$ cd ~

$ source .sbai/bin/activate
```

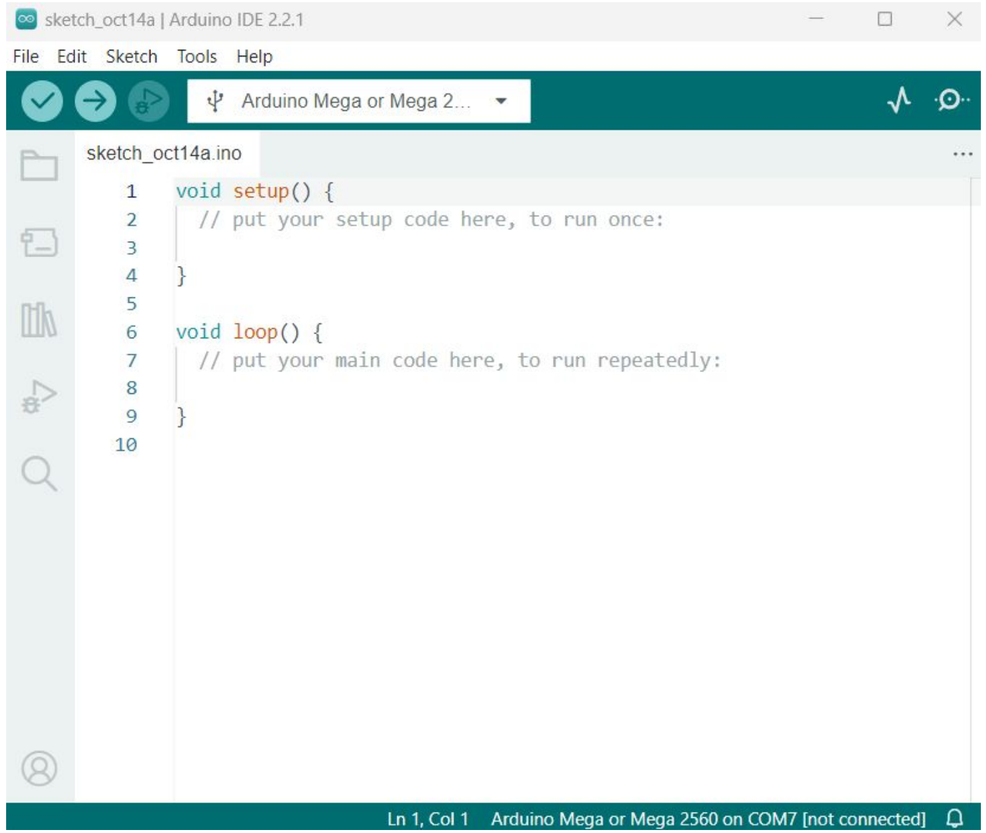


EXP I: Hello World!!

- **2. Abrir a Arduino IDE por meio do ambiente virtual .sbai:**
  - Para isso executem os seguintes comandos:

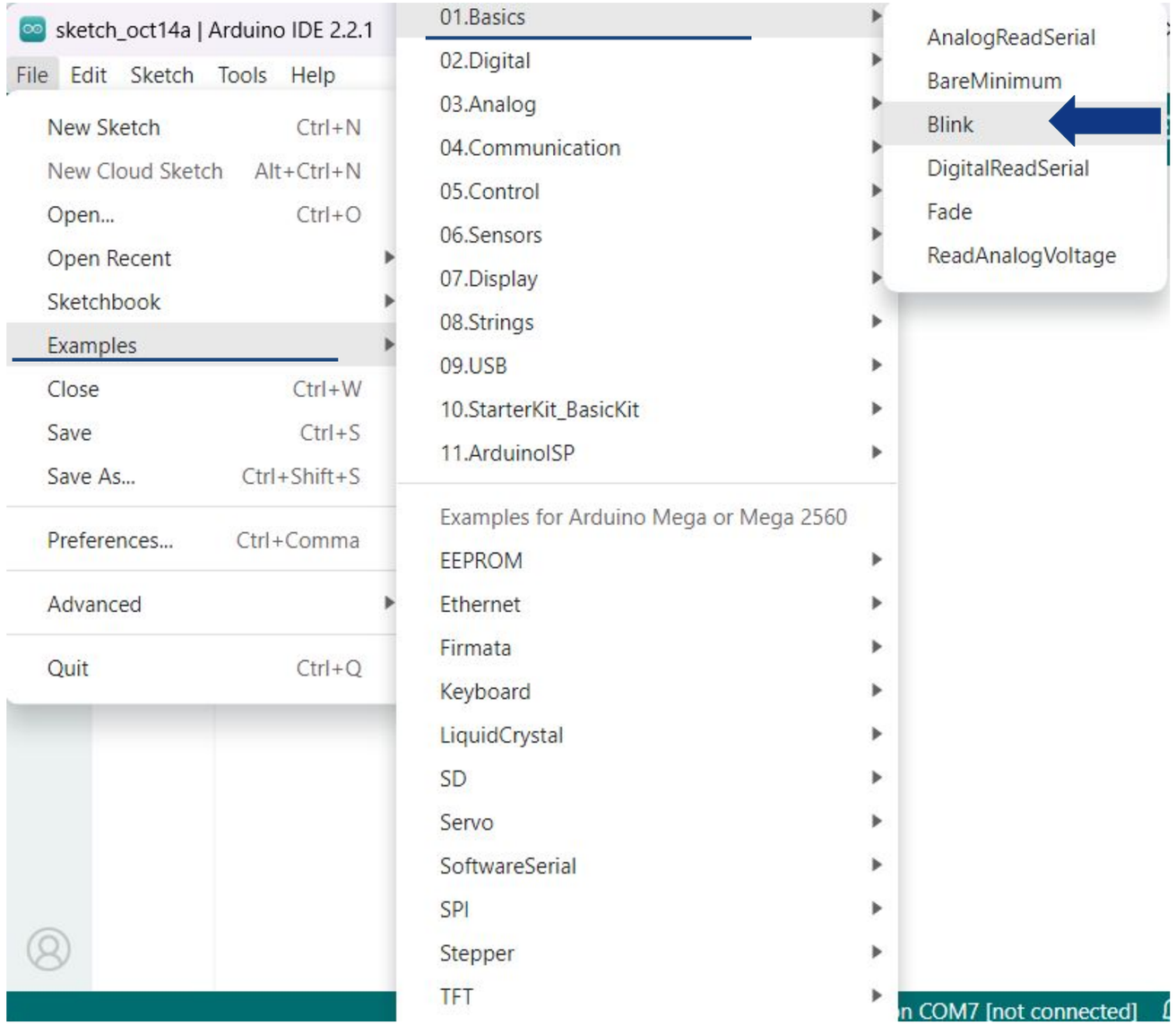
```
(.sbai) $ cd Downloads/arduino

(.sbai) $ ./arduino-ide
```



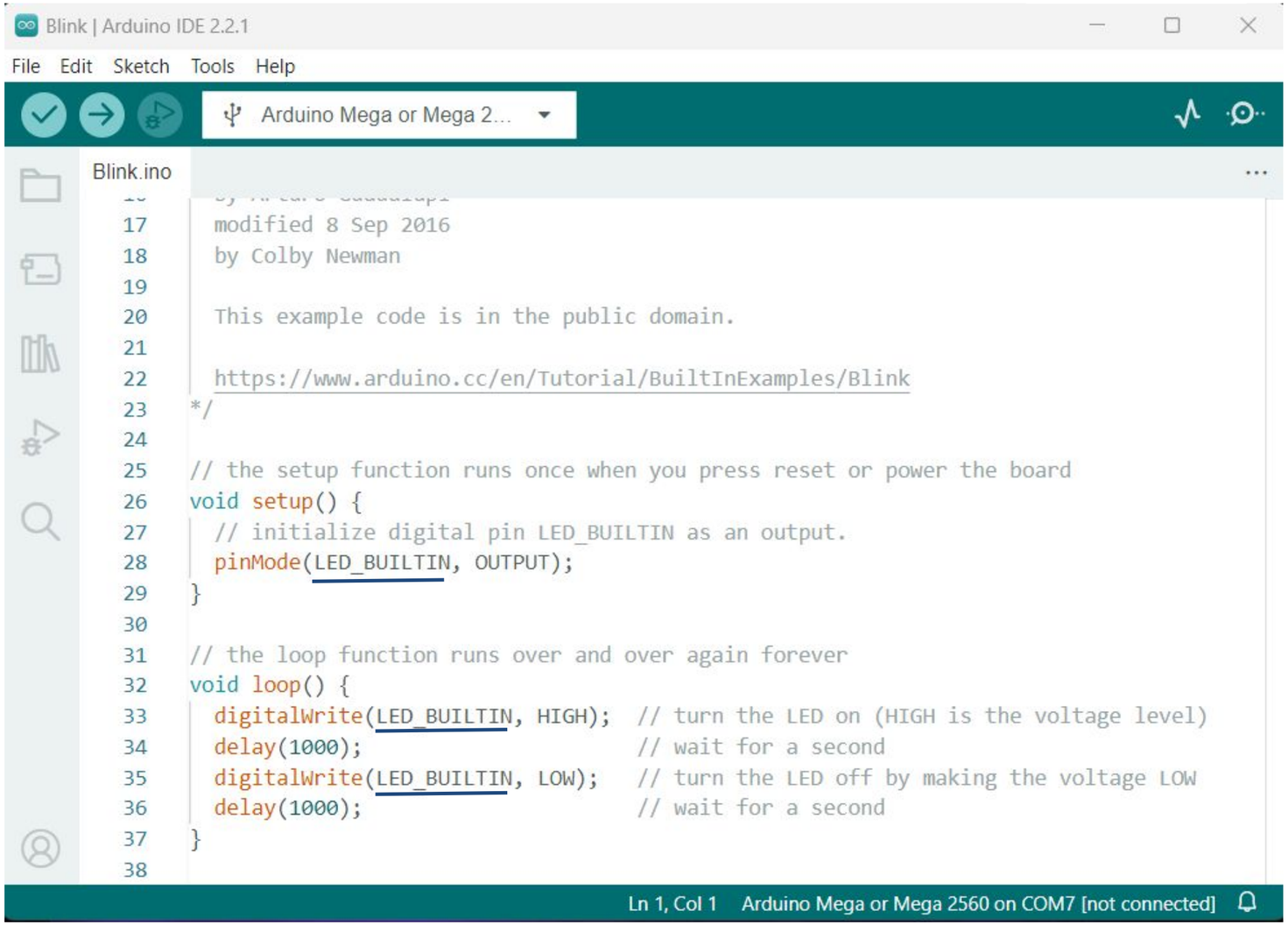
EXP I: Hello World!!

- **3. Abrir o exemplo *Blink*:**
  - File > Examples > Basics > Blink



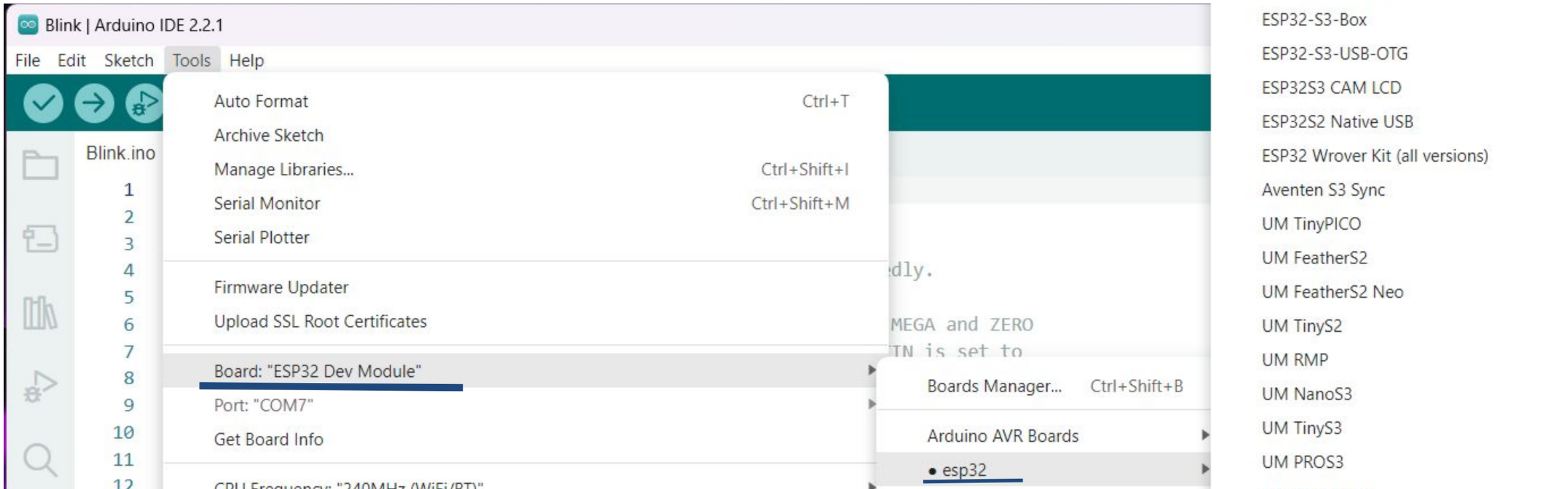
EXP I: Hello World!!

- **4. Altere a pinagem:**
  - LED_BUILTIN => 2



EXP I: Hello World!!

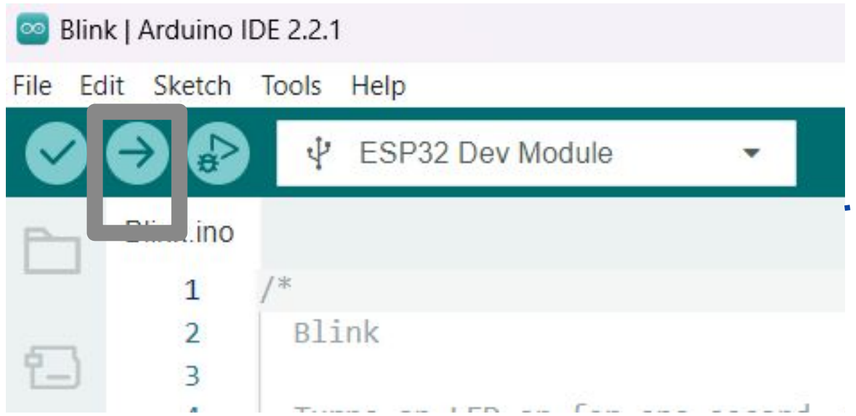
- **5. Selecione a placa ESP32:**
  - Tools > Board > esp32 > ESP32 Dev Module

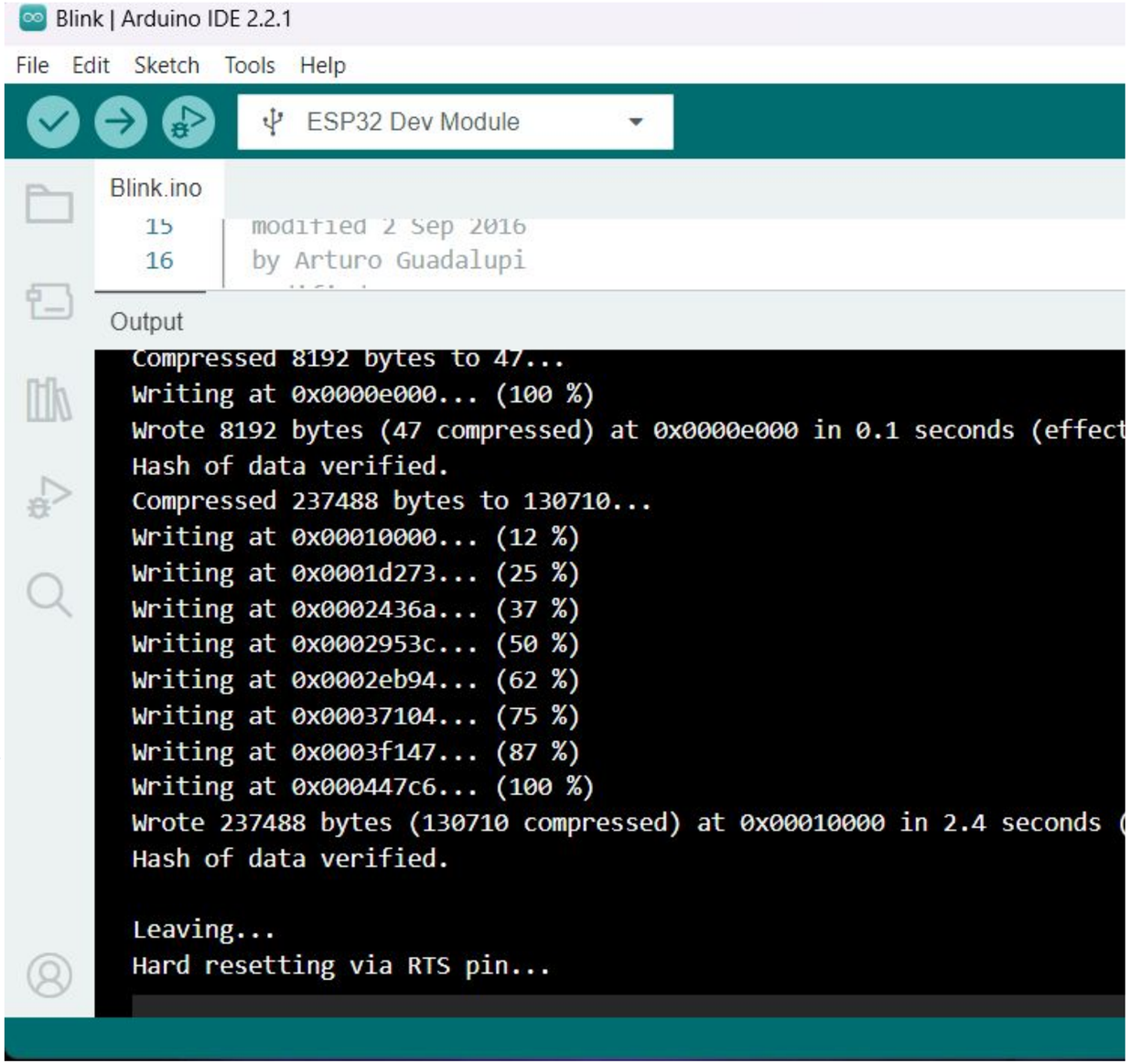


EXP I: Hello World!!

- **6. Selecione a porta serial:**
  - Tools > Port > ***/dev/ttyACM0***

- **7. Clique em Upload:**





EXP II: Classif. de Imagens

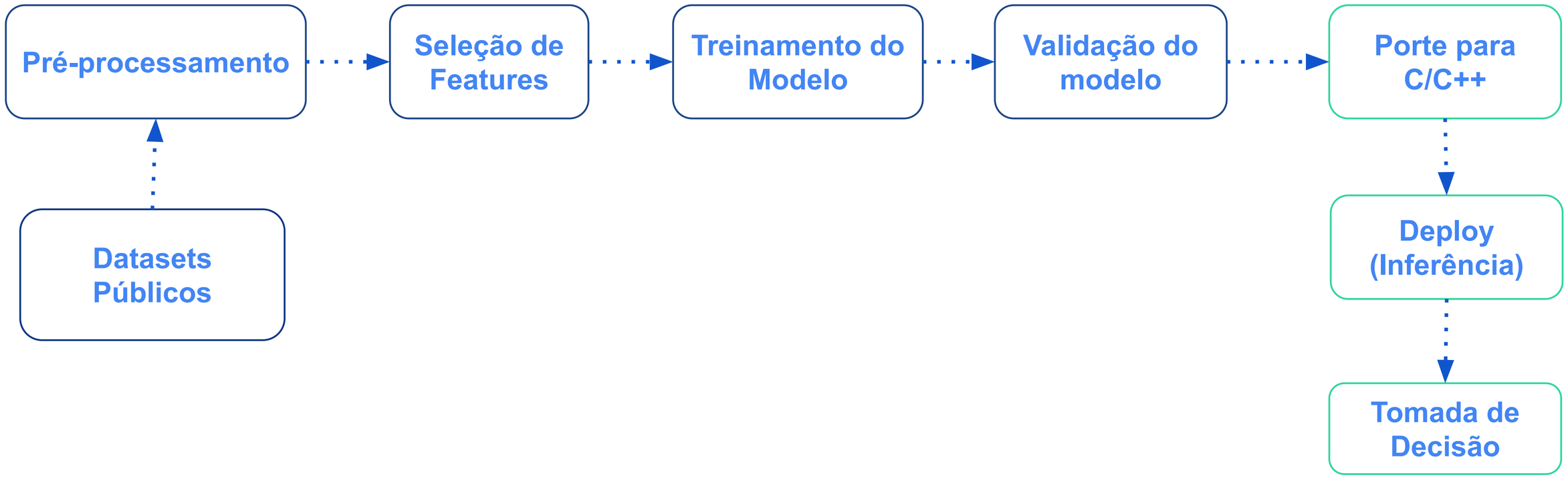
- **MICRO ML GENERATOR**
  - **Projeto opensource**
  - **Foca no porte de Algorítmos de ML Clássicos**
  - **Suporta o porte para:**
    - Decision Tree
    - RandomForest
    - XGBoost
    - GaussianNB
    - Support Vector Machines (SVC and OneClassSVM)
    - Relevant Vector Machines
    - PCA

Eloquent Arduino

Level up your TinyML skills

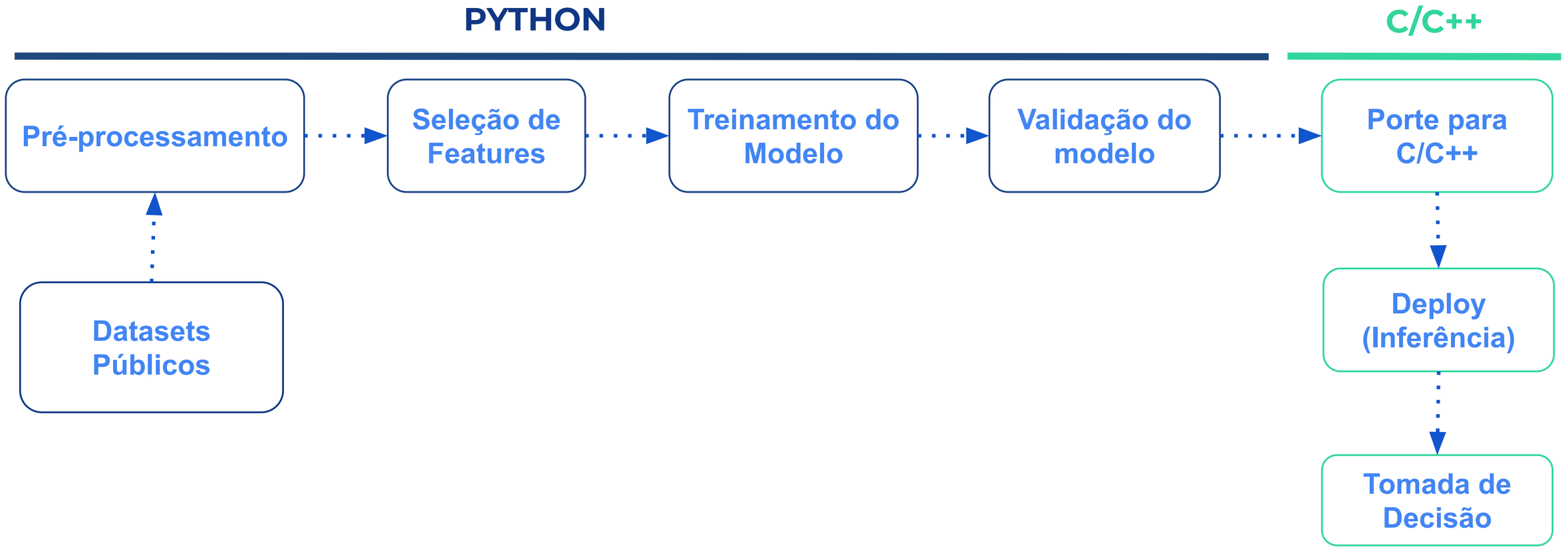
EXP II: Classif. de Imagens

- **Fluxo TinyML:**



EXP II: Classif. de Imagens

- **Fluxo TinyML**



EXP II: Classif. de Imagens

- **Objetivo:**
  - Obter familiaridade com a lib **MicroML Generator;**
  - Realizar a classificação de Imagens no MCU.

Iris - Setosa

Iris - Versicolor

Iris - Virginica

EXP II: Classif. de Imagens

- **Iris dataset:**
  - Classes: 3;
  - Amostras p/ classes: 50;
  - Tamanho total: 150;
  - Dimensionalidade: 4.

# SepalLeng...	# SepalWidt...	# PetalLengt...	# PetalWidt...	Species
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5.0	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa

Iris Dataset

EXP II: Classif. de Imagens

- **Iris dataset:**
  - Classes: 3;
  - Amostras p/ classes: 50;
  - Tamanho total: 150;
  - Dimensionalidade: 4.

X.p1 + X.p2 + X.p3 + ... = y

# SepalLeng...	# SepalWidt...	# PetalLengt...	# PetalWidt...	A Species
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5.0	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa

EXP I: Classif. de Imagens

- **1. Acesse o diretório home,**
  - Crie um arq. chamado **iris_classifier.py**
  - Dentro do arquivo digite:
  - Execute:

(.sbai) $ python iris_classifier.py

```
# Import ML Libs
from micromlgen import port
from sklearn import metrics
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression

# Main program
if __name__ == '__main__':

    # Load iris dataset:
    X, y = load_iris(return_X_y=True)
    print("Shape of X:", X.shape, "\tSize of y:", len(y))
    print("X:", X)
    print("\ny:", y)
```

- **2. Treine o modelo**
  - Continue digitando:

```
(.sbai) $ python iris_classifier.py
```

```
18      # Splitting into train and test
19      X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.25)
20
21      # Train a Random Forest Model
22      model = RandomForestClassifier(n_estimators=10)
23      model.fit(X_train, y_train)
24
25      # --- Performance in Training ---
26      training_prediction = model.predict(X_train)
27      print(training_prediction)
```

- **3. Avalie a performance do modelo na base de TREINO**
  - Continue digitando:

(.sbai) $ python iris_classifier.py

```
25      # --- Performance in Training ---
26      training_prediction = model.predict(X_train)
27      print(training_prediction)
28
29      print("\nPrecision, Recall, Confusion matrix, in training\n")
30      # Precision Recall scores
31      print(metrics.classification_report(y_train, training_prediction, digits=3))
32      # Confusion matrix
33      print(metrics.confusion_matrix(y_train, training_prediction))
34      # ------------------------------------
```

- **4. Avalie a performance do modelo na base de TESTE**
  - Continue digitando:

(.sbai) $ python iris_classifier.py

```
36      # --- Performance in Test ---
37      test_prediction = model.predict(X_test)
38      print(training_prediction)
39
40      print("\nPrecision, Recall, Confusion matrix, in testing\n")
41      # Precision Recall scores
42      print(metrics.classification_report(y_test, test_prediction, digits=3))
43      # Confusion matrix
44      print(metrics.confusion_matrix(y_test, test_prediction))
45      # ------------------------------------
```

EXP I: Classif. de Imagens

- **5. Teste amostras isoladas no modelo**
  - Continue digitando:

(.sbai) $ python iris_classifier.py

```
48    # --- Test classification in randomic samples ---
49    # Sample of a Setosa
50    print('The accuracy of model is:', model.predict([[5.1, 3.5, 1.4, 0.2]]))
51
52    # Sample of a Versicolor
53    print('The accuracy of model is:', model.predict([[6.2, 2.2, 4.5,1.5]]))
54
55    # Sample of a Virginica
56    print('The accuracy of model is', model.predict([[6.1, 3.0, 4.9,1.8]]))
57
```

- **6. Porte o modelo para C/C++**
  - Continue digitando:

```
(.sbai) $ python iris_classifier.py
```

```
59      # --- Performance in Training ---
60      c_code = port(model)
61      with open('iris_classifier.h', 'w') as file:
62          file.write(c_code)
63
```

EXP I: Classif. de Imagens

- **6. Porte o modelo para C/C++**
  - Crie um novo sketch no Arduino IDE;
  - Copie a função **predict** do **iris.classifier.h**

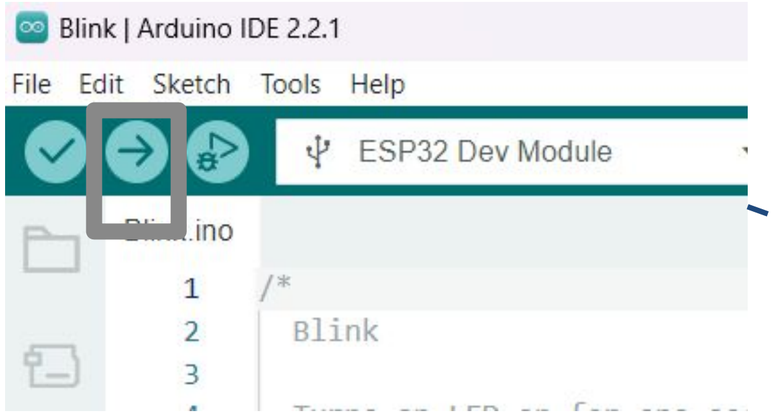
    gerado p/ o novo sketch do Arduino IDE

iris_classifier.ino

```
1
2   /**
3   * Predict class for features vector
4   */
5   int predict(float *x) {…
391 }
392
393 void setup()
394 {
395   Serial.begin(115200);
396 }
397
398 void loop() {
399   float X_1[] = {5.1, 3.5, 1.4, 0.2};
400   int result = predict(X_1);
401
402   Serial.print("Result of predict with input X1:");
403   Serial.println(result);
404   delay(2000);
405
406   float X_2[] = {6.2, 2.2, 4.5,1.5};
407   result = predict(X_2);
408   Serial.print("Result of predict with input X2:");
409   Serial.println(result);
410   delay(2000);
411
412   float X_3[] = {6.1, 3.0, 4.9,1.8};
413   result = predict(X_3);
414   Serial.print("Result of predict with input X3:");
415   Serial.println(result);
416   delay(2000);
417 }
```

EXP I: Classif. de Imagens

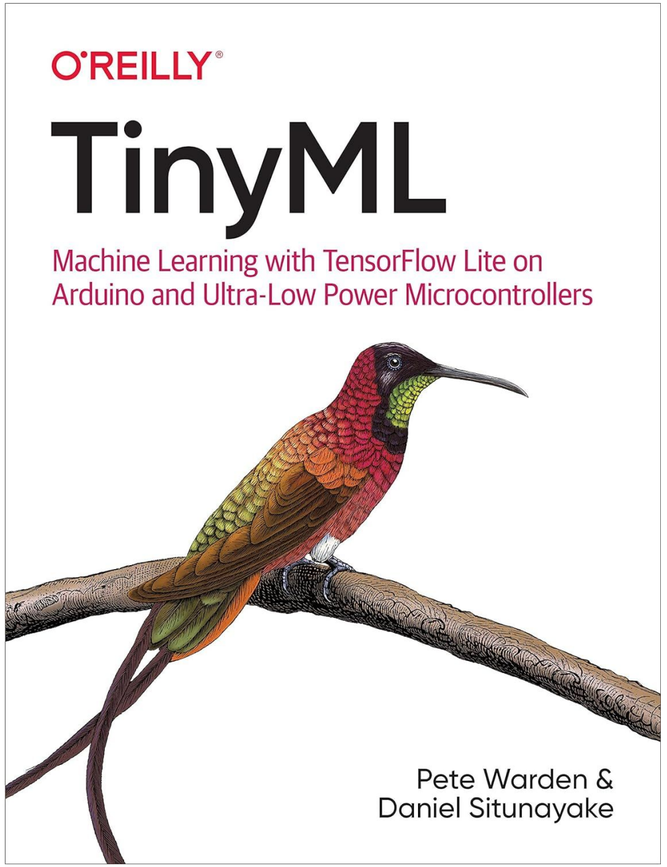
- **7. Clique em Upload:**

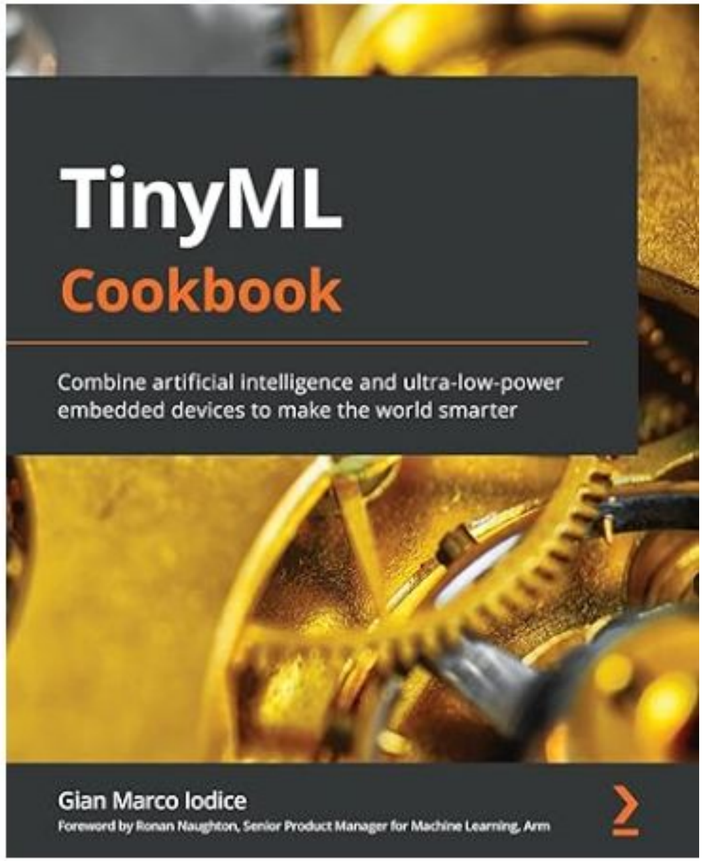


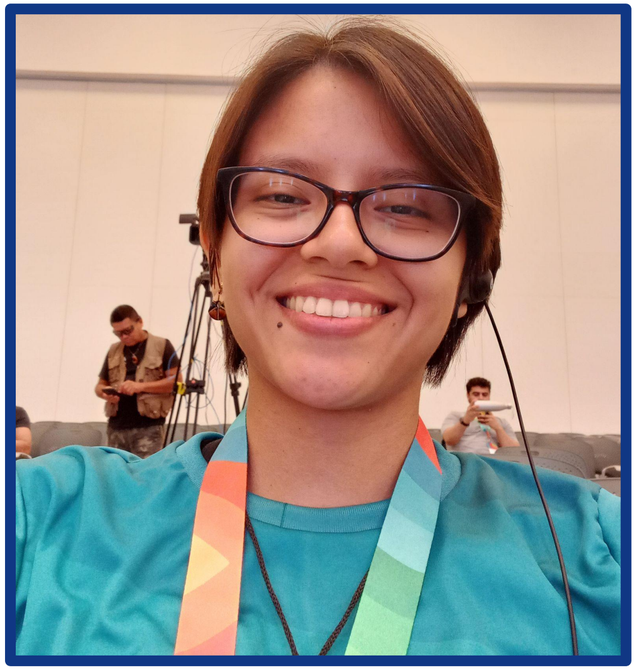

```
Result of predict with input X1:0
Result of predict with input X2:1
Result of predict with input X3:2
```

Próximos passos

- **Livros**
- **TinyML Foundation**
- **LinkedIn TinyML**







Obrigada!

Perguntas?

CONTATO:

- **Email:** lahis.almeida@indt.org.br
- **LinkedIn**
- **Material do curso**


INDT
INSTITUTO DE DESENVOLVIMENTO
TECNOLÓGICO