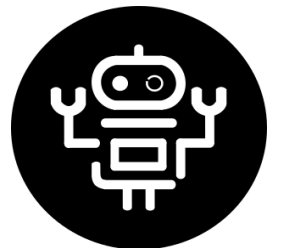
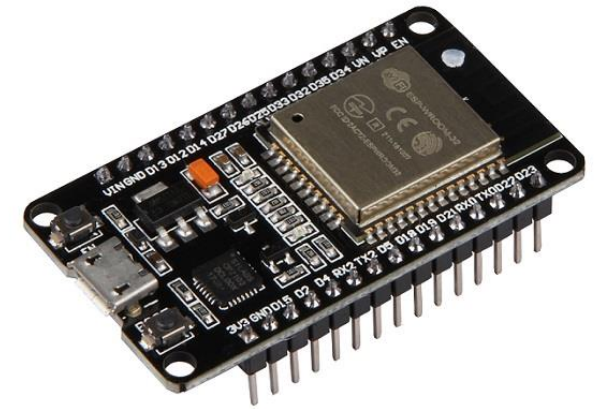
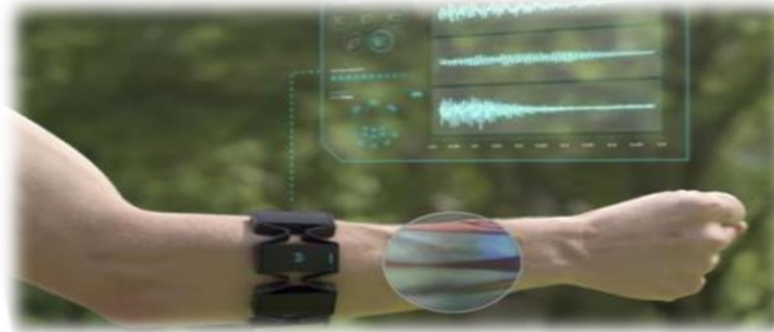
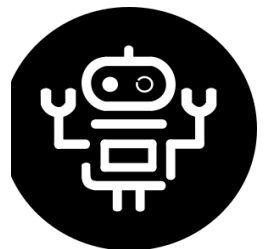
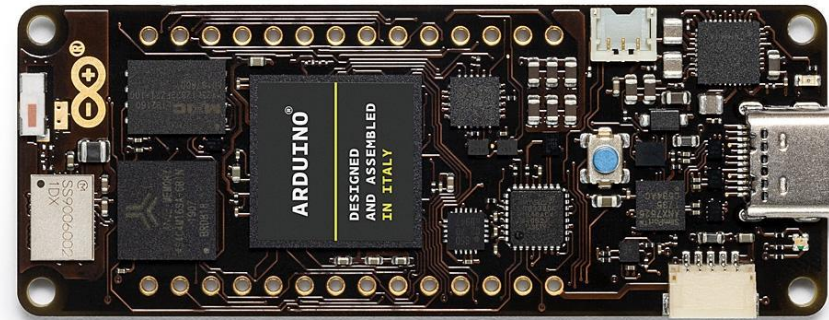
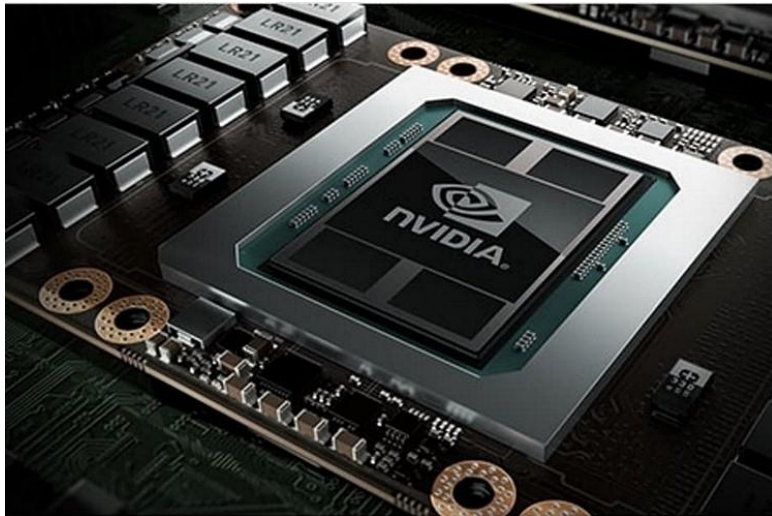


REDES NEURONALES ARTIFICIALES CON TENSORFLOW Y ESP32.

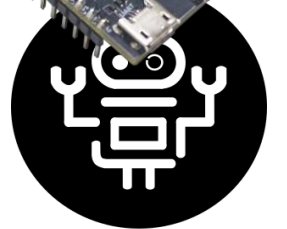
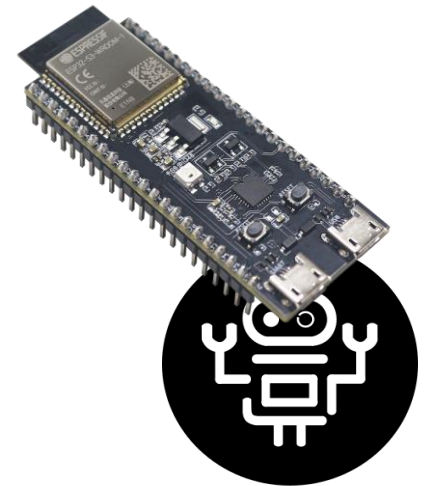
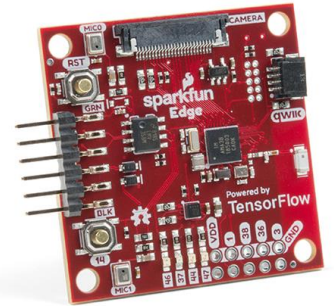
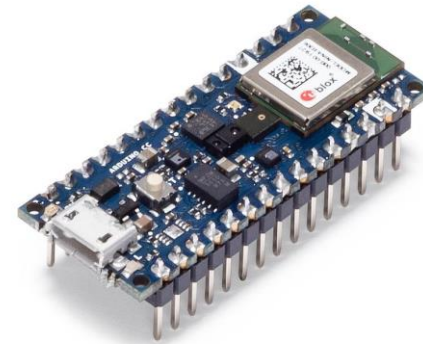
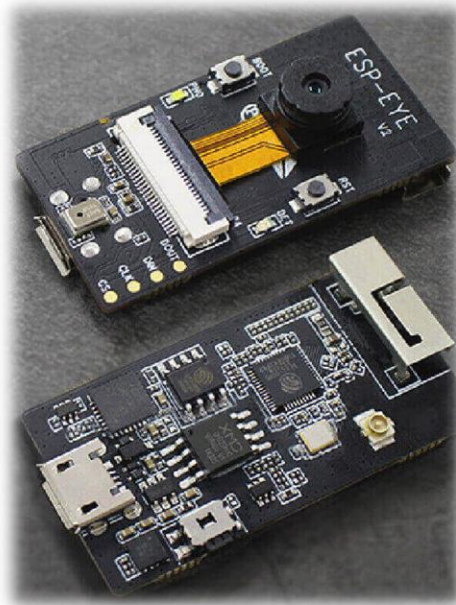
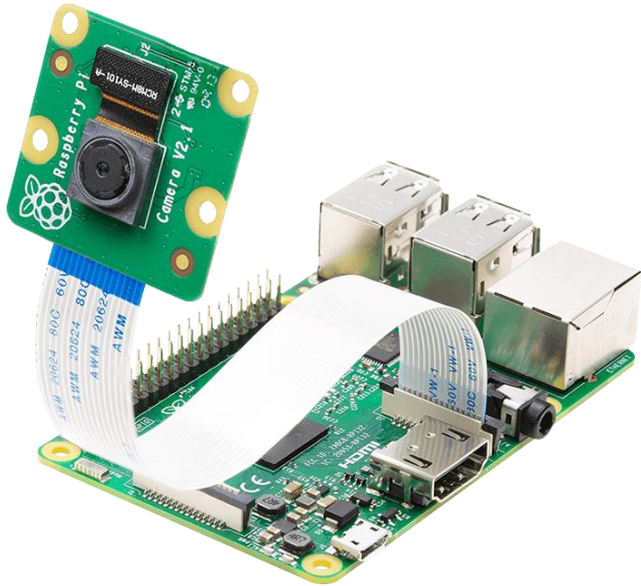


MICROCONTROLADORES PARA APRENDIZAJE AUTOMÁTICO

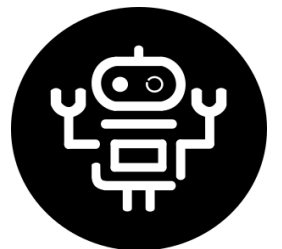
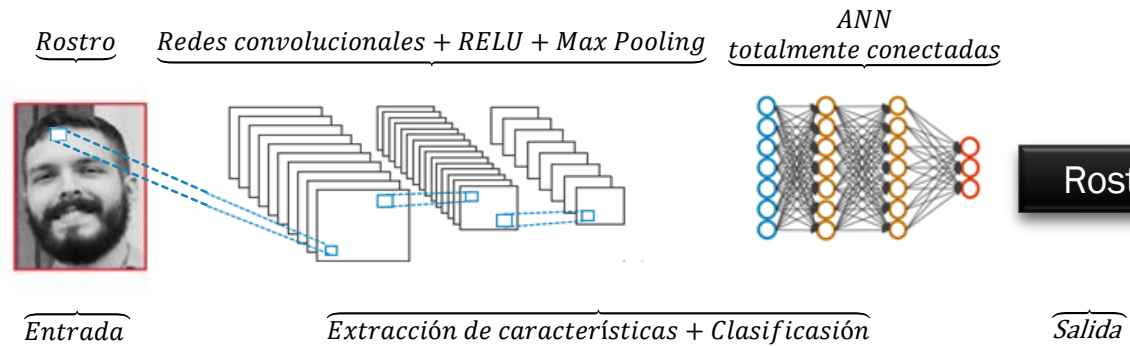
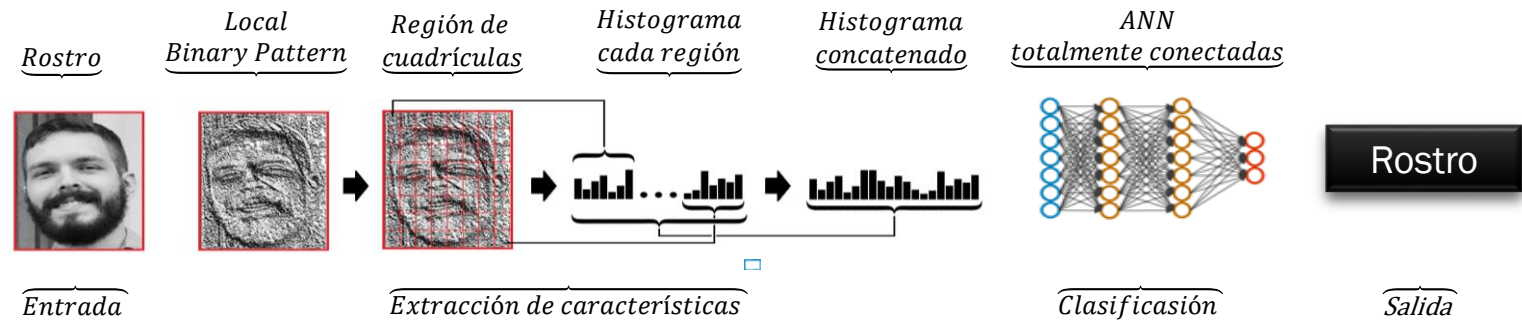
- El aprendizaje profundo y aprendizaje automático. Siempre se han asociado con computadoras grandes con CPU y GPU rápidas, gran tamaño de RAM o algoritmos en ejecución en la nube.
- Sin embargo, imagina realizar Machine Learning en un microcontrolador alimentado por una sola batería. Puede pensar que es imposible, pero con la tecnología actual, lo imposible ahora es posible con los microcontroladores.



MICROCONTROLADORES PARA APRENDIZAJE AUTOMÁTICO



MACHINE LEARNING VS DEEP LEARNING

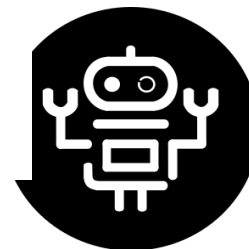


INSTALACIÓN PYTHON Y MÓDULOS NECESARIOS

<https://www.tensorflow.org>

Empresas que usan TensorFlow

Ver todos los casos de éxito →



www.roboticoss.com

INSTALACIÓN PYTHON Y MÓDULOS NECESARIOS

<https://www.tensorflow.org/install?hl=es>

Instala TensorFlow 2

TensorFlow se probó con los siguientes sistemas de 64 bits y es compatible con ellos:

- Python 3.6 a 3.9
- Ubuntu 16.04 o versiones posteriores
- Windows 7 o versiones posteriores (con [C++ redistribuible](#))
- macOS 10.12.6 (Sierra) o versiones posteriores (no tiene compatibilidad con GPU)

Descarga un paquete

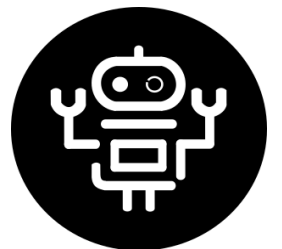
Instala TensorFlow con el administrador de paquetes *pip* de Python.

★ Los paquetes de TensorFlow 2 requieren una versión de **pip** posterior a 19.0 (o posterior a 20.3 para macOS).

```
# Requires the latest pip
$ pip install --upgrade pip
```

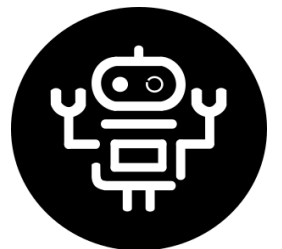
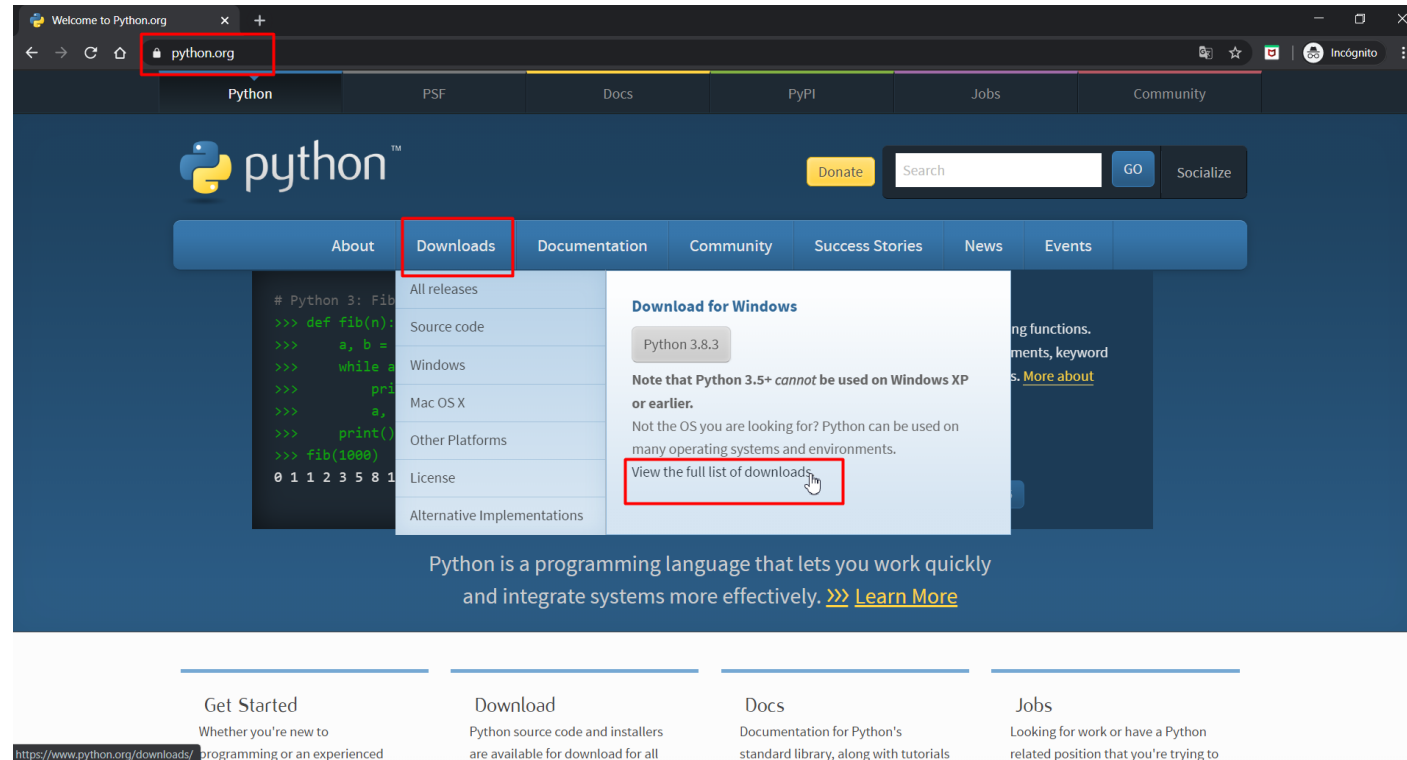
```
# Current stable release for CPU and GPU
$ pip install tensorflow
```

```
# Or try the preview build (unstable)
$ pip install tf-nightly
```



INSTALACIÓN PYTHON Y MÓDULOS NECESARIOS

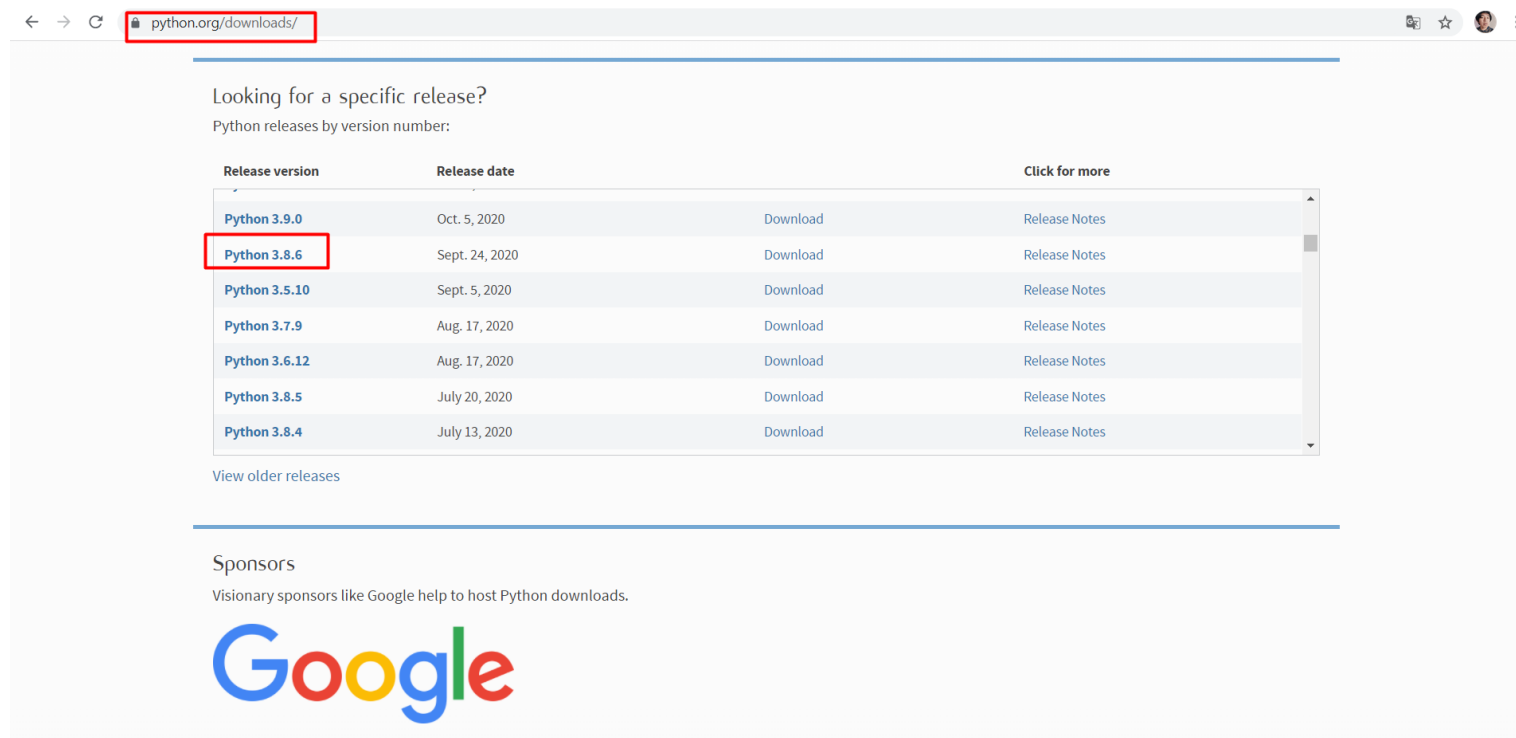
<https://www.python.org/>



www.roboticoss.com

INSTALACIÓN PYTHON Y MÓDULOS NECESARIOS

<https://www.python.org/downloads/>

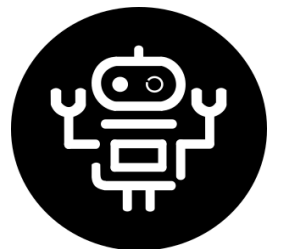


The screenshot shows the Python.org downloads page. The browser's address bar is highlighted with a red box, showing the URL `python.org/downloads/`. The page content includes a heading "Looking for a specific release?" and a subheading "Python releases by version number:". Below this is a table with three columns: "Release version", "Release date", and "Click for more". The table lists several Python versions, with "Python 3.8.6" highlighted by a red box. Below the table is a link "View older releases". At the bottom of the page, there is a "Sponsors" section with the text "Visionary sponsors like Google help to host Python downloads." and the Google logo.

Release version	Release date	Click for more
Python 3.9.0	Oct. 5, 2020	Download Release Notes
Python 3.8.6	Sept. 24, 2020	Download Release Notes
Python 3.5.10	Sept. 5, 2020	Download Release Notes
Python 3.7.9	Aug. 17, 2020	Download Release Notes
Python 3.6.12	Aug. 17, 2020	Download Release Notes
Python 3.8.5	July 20, 2020	Download Release Notes
Python 3.8.4	July 13, 2020	Download Release Notes

[View older releases](#)

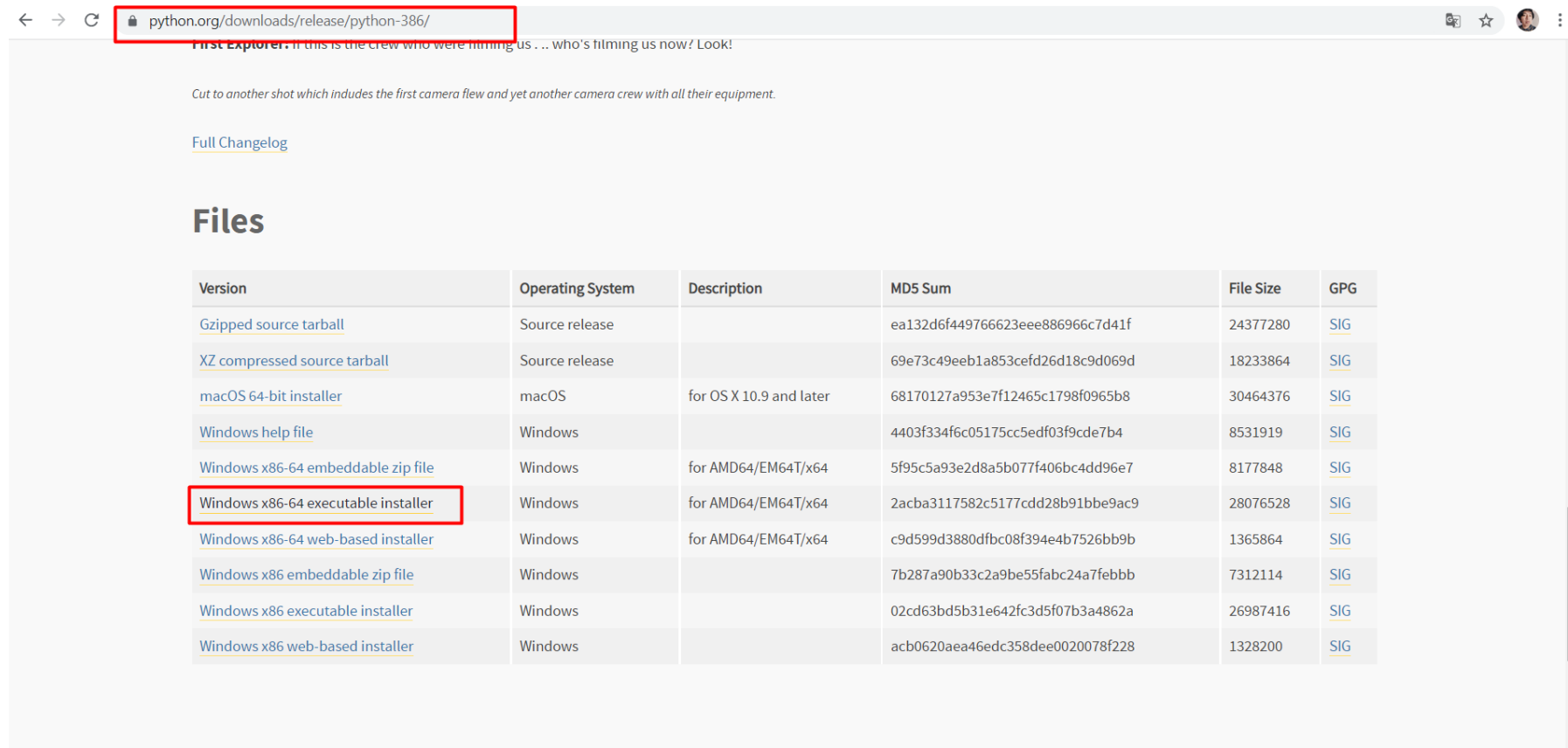
Sponsors
Visionary sponsors like Google help to host Python downloads.



www.roboticoss.com

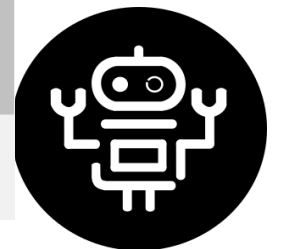
INSTALACIÓN PYTHON Y MÓDULOS NECESARIOS

<https://www.python.org/downloads/release/python-386/>

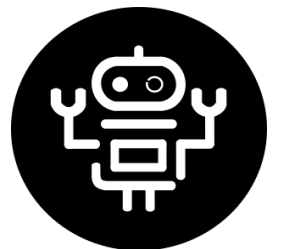
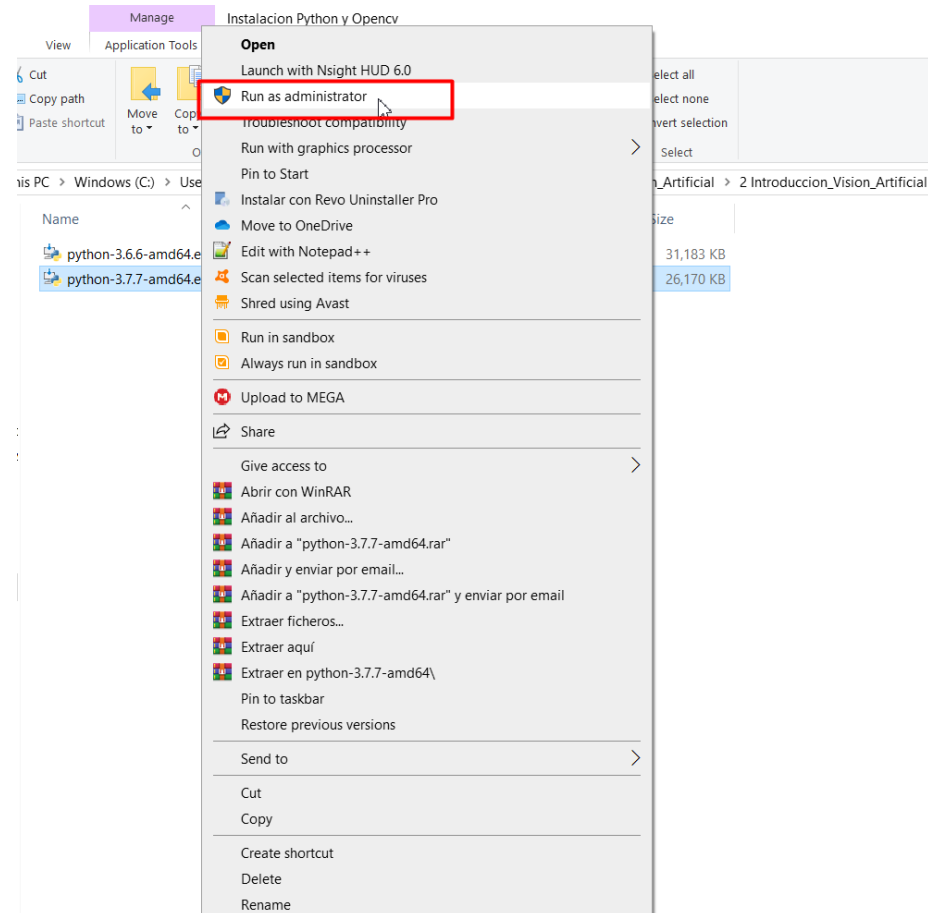


The screenshot shows the Python 3.8.6 download page. The browser's address bar is highlighted with a red box, showing the URL `python.org/downloads/release/python-386/`. Below the address bar, there is a section titled "Files" which contains a table of download links. The table has six columns: Version, Operating System, Description, MD5 Sum, File Size, and GPG. The row for "Windows x86-64 executable installer" is highlighted with a red box.

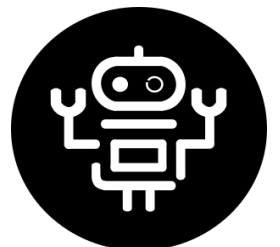
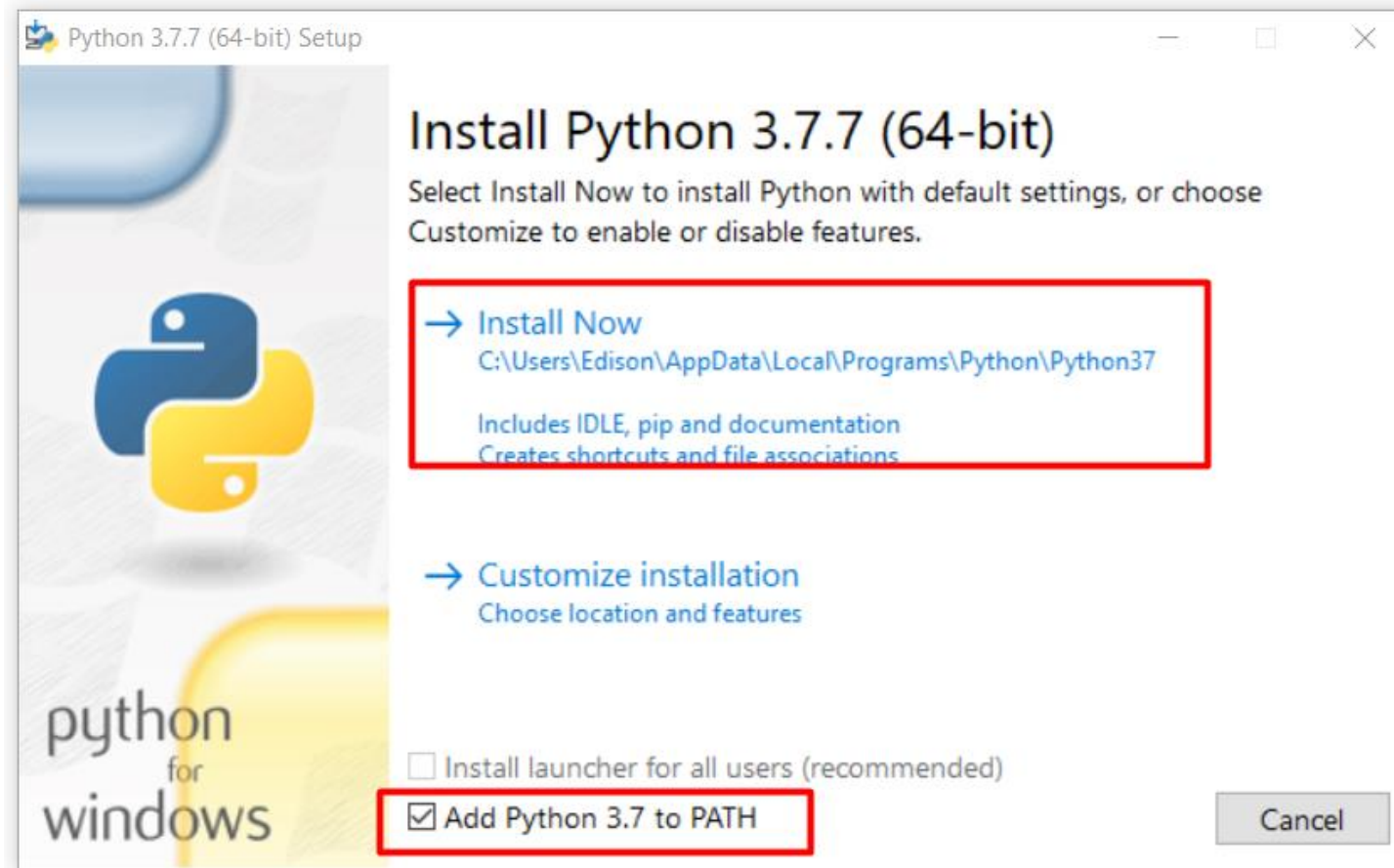
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		ea132d6f449766623eee886966c7d41f	24377280	SIG
XZ compressed source tarball	Source release		69e73c49eeb1a853cefd26d18c9d069d	18233864	SIG
macOS 64-bit installer	macOS	for OS X 10.9 and later	68170127a953e7f12465c1798f0965b8	30464376	SIG
Windows help file	Windows		4403f334f6c05175cc5edf03f9cde7b4	8531919	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	5f95c5a93e2d8a5b077f406bc4dd96e7	8177848	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	2acba3117582c5177cdd28b91bbe9ac9	28076528	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	c9d599d3880dfbc08f394e4b7526bb9b	1365864	SIG
Windows x86 embeddable zip file	Windows		7b287a90b33c2a9be55fab24a7febbb	7312114	SIG
Windows x86 executable installer	Windows		02cd63bd5b31e642fc3d5f07b3a4862a	26987416	SIG
Windows x86 web-based installer	Windows		acb0620aea46edc358dee0020078f228	1328200	SIG



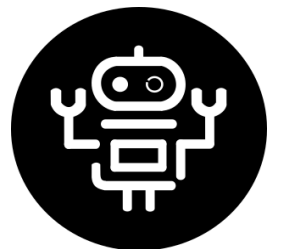
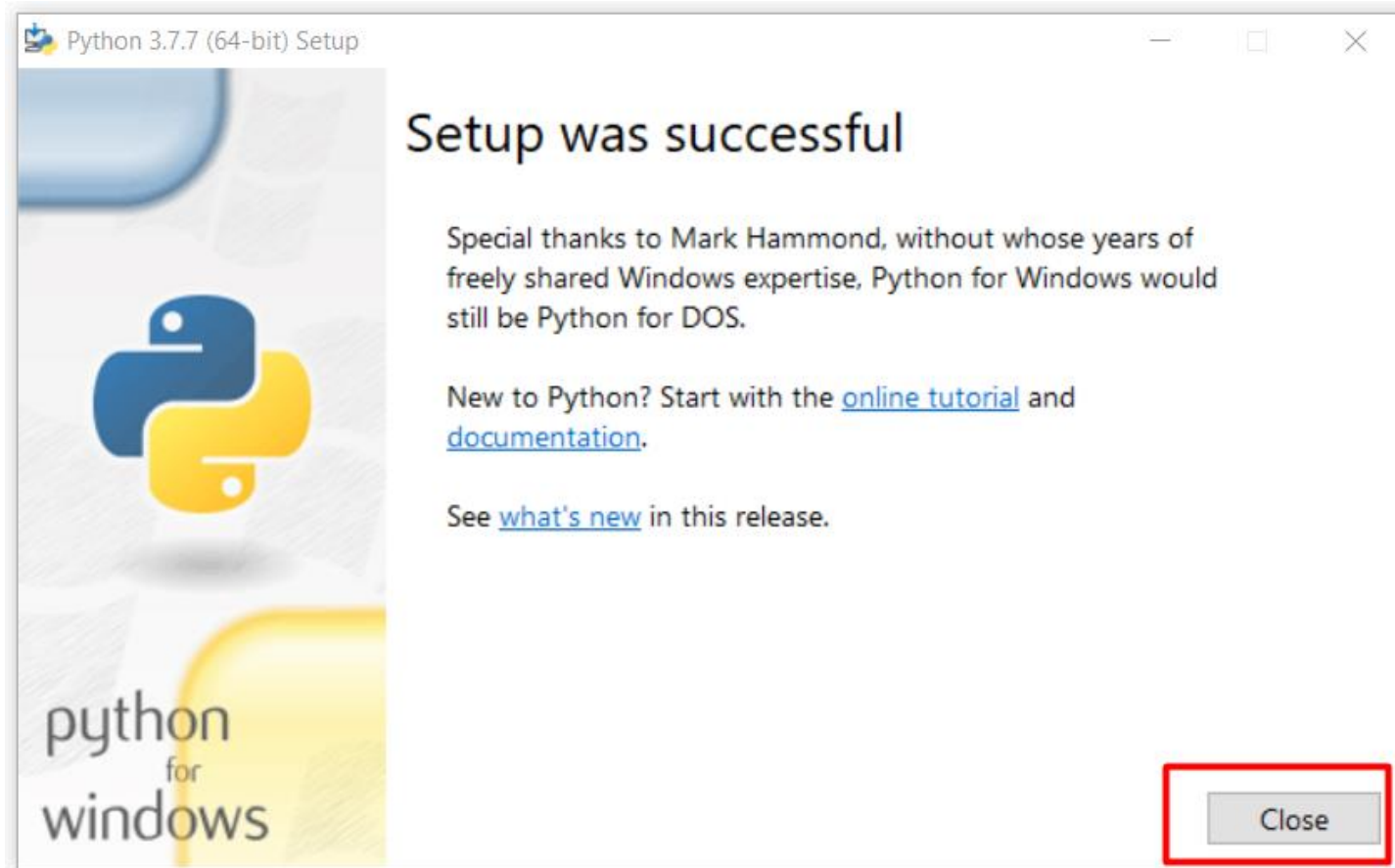
INSTALACIÓN PYTHON



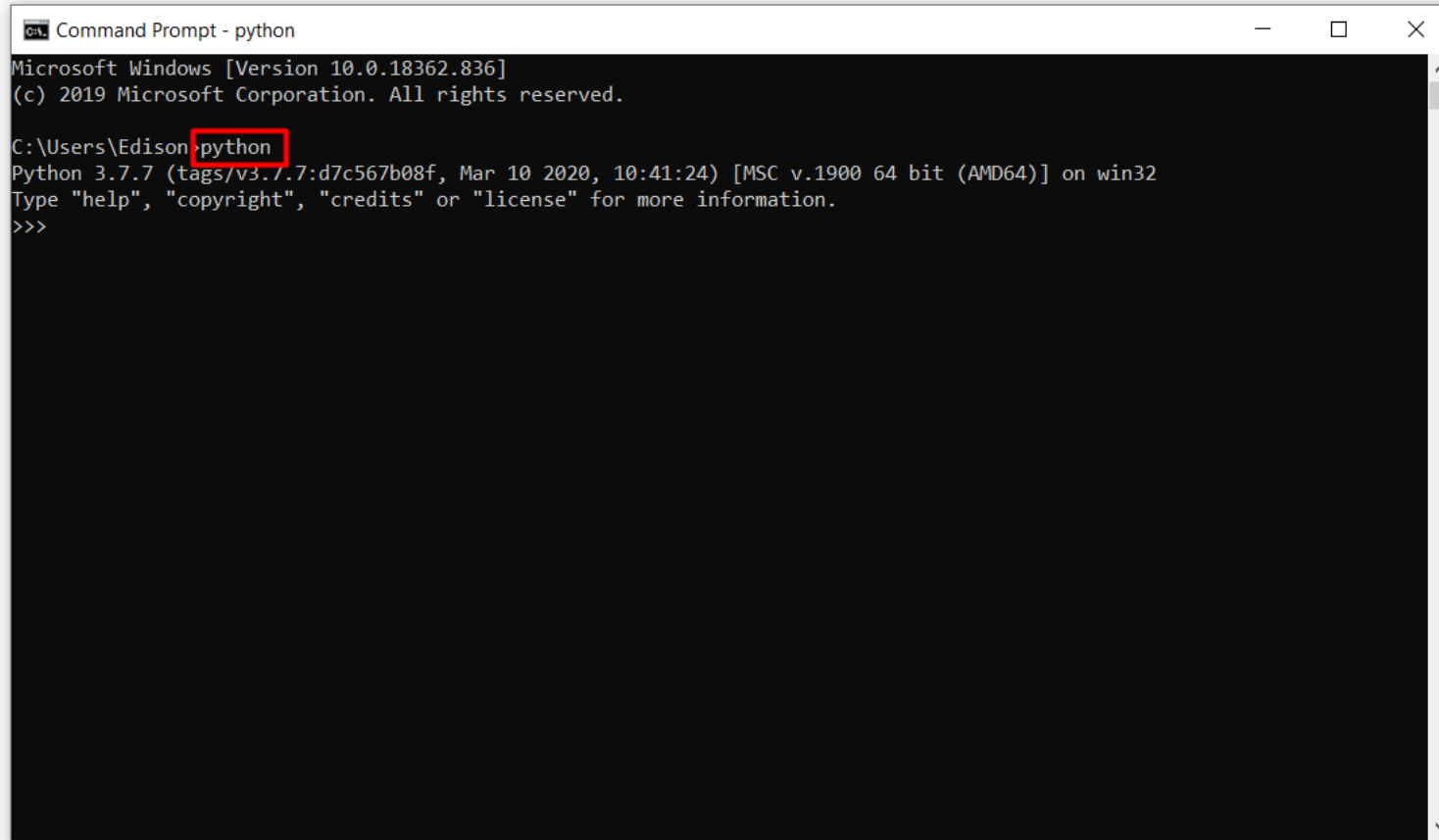
INSTALACIÓN PYTHON



INSTALACIÓN PYTHON

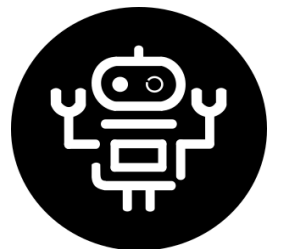


INSTALACIÓN PYTHON

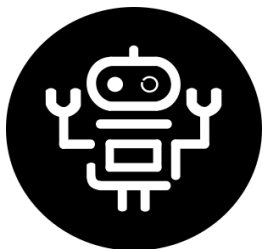
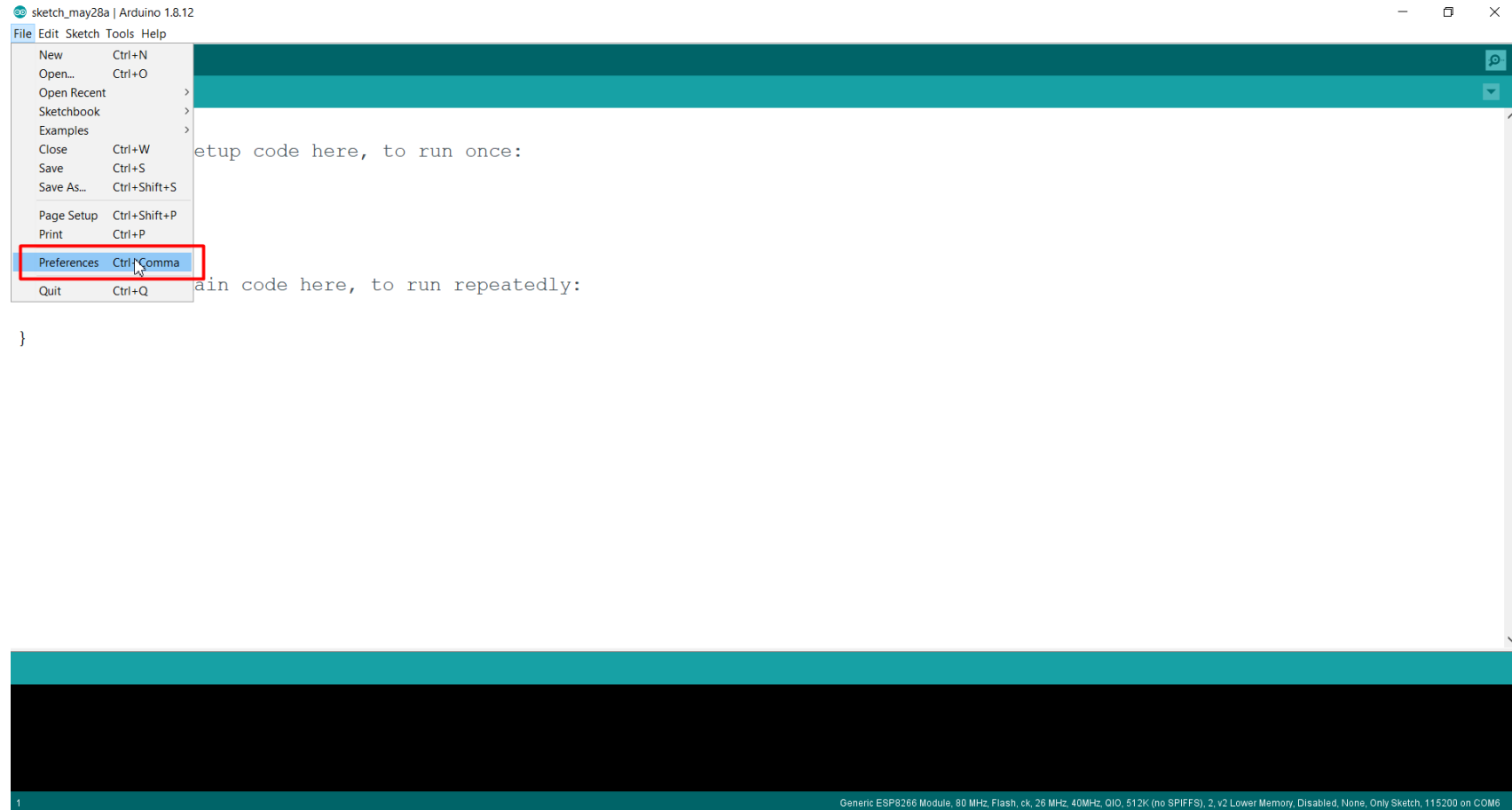


```
Command Prompt - python
Microsoft Windows [Version 10.0.18362.836]
(c) 2019 Microsoft Corporation. All rights reserved.

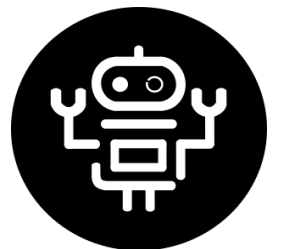
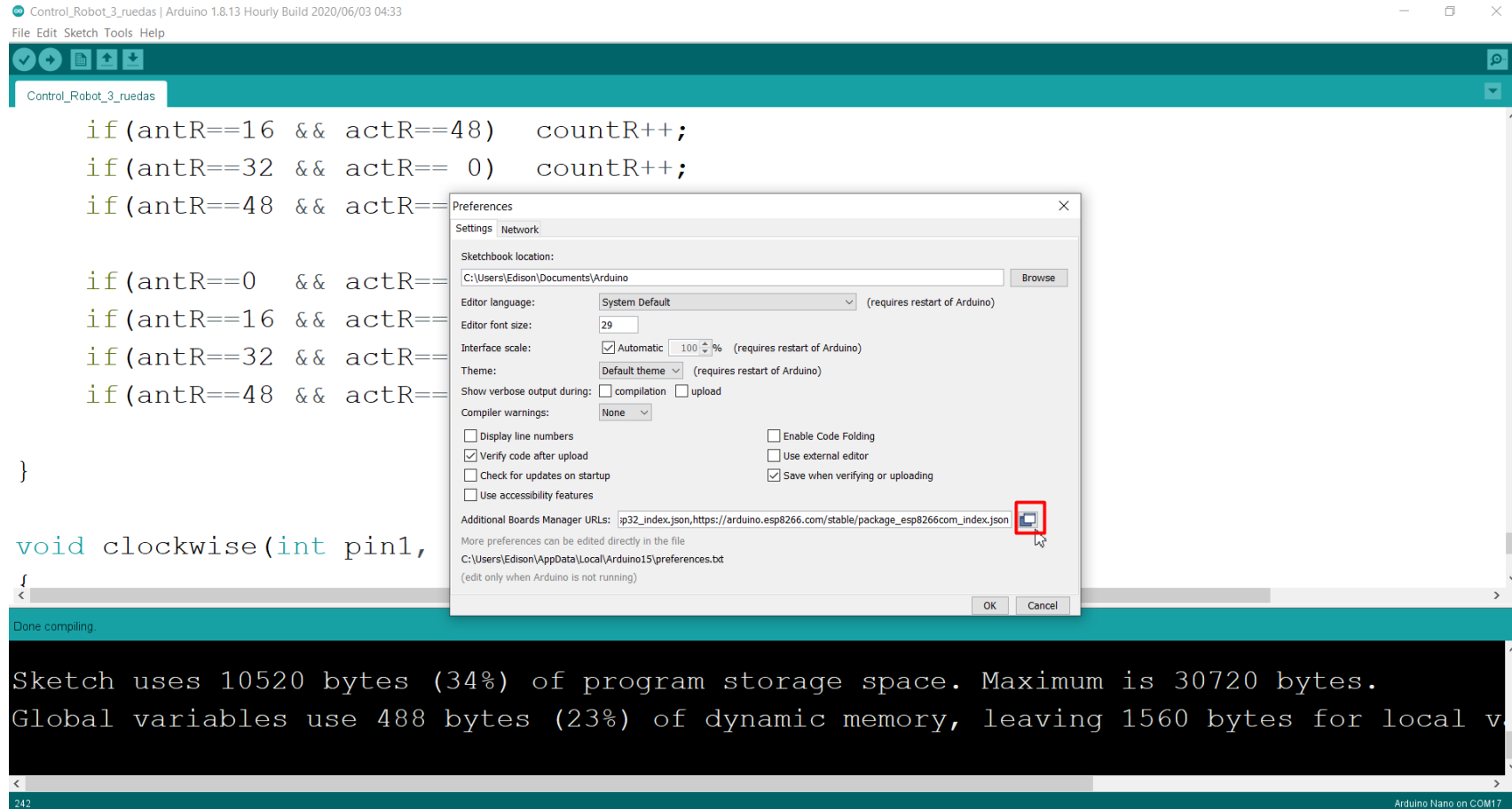
C:\Users\Edison>python
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```



INSTALACIÓN DE LA TARJETA ESP32 EN ARDUINO IDE

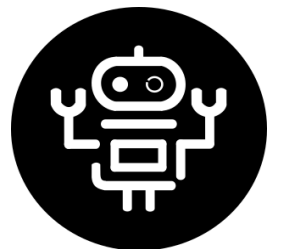
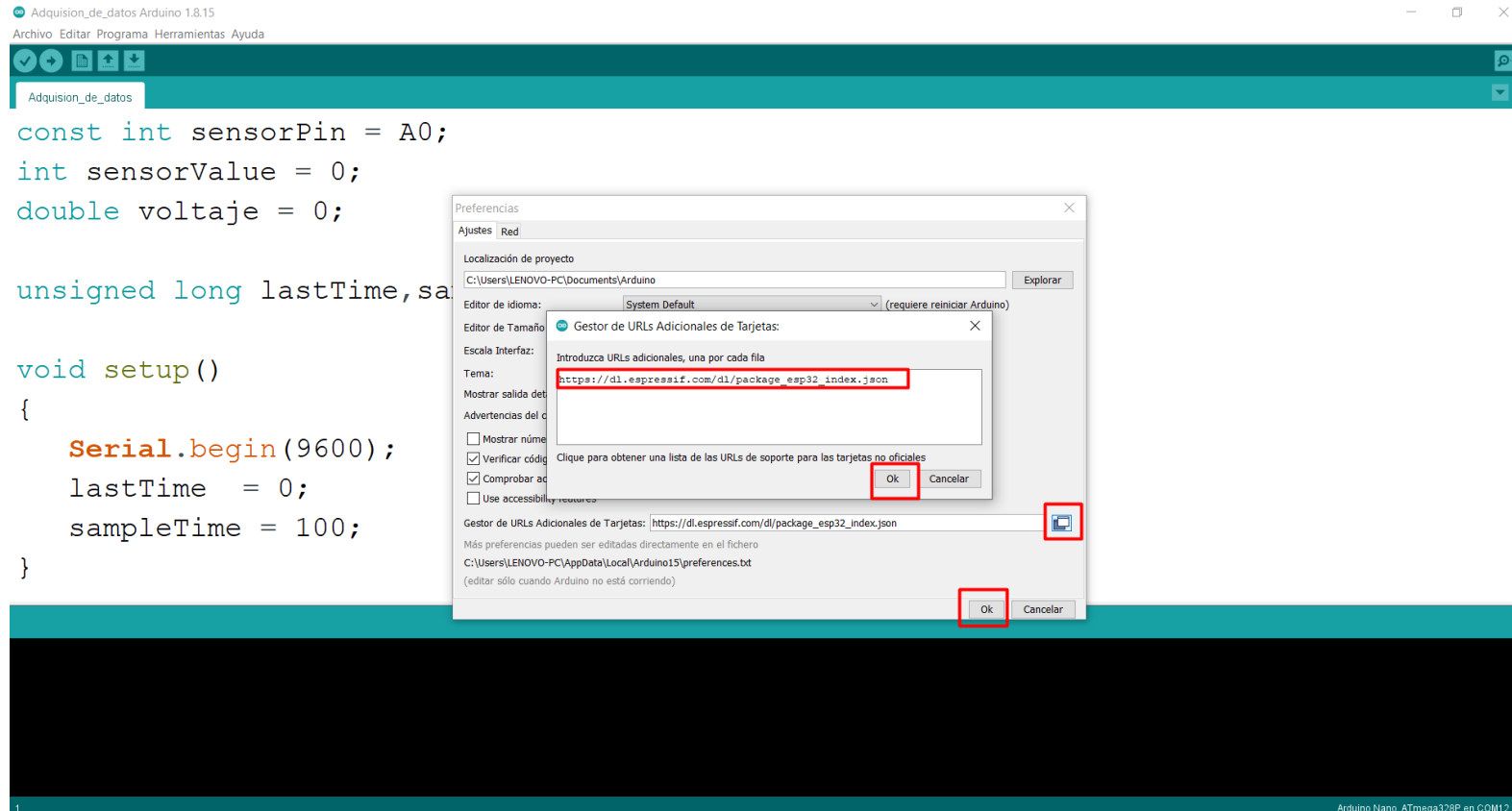


INSTALACIÓN DE LA TARJETA ESP32 EN ARDUINO IDE



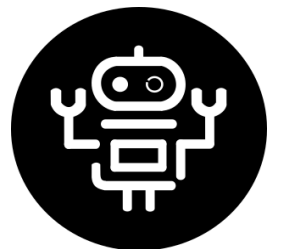
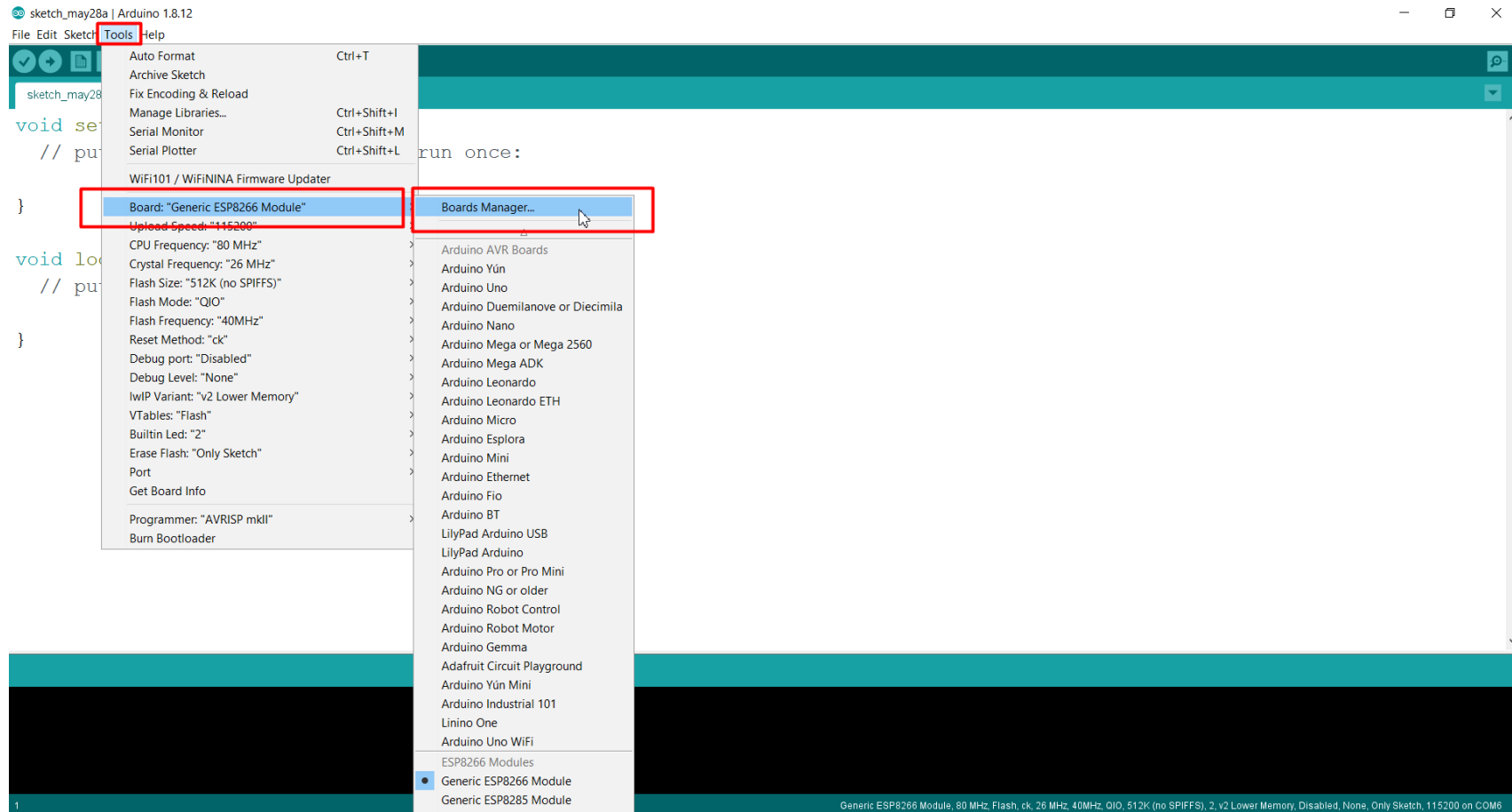
INSTALACIÓN DE LA TARJETA ESP32 EN ARDUINO IDE

https://dl.espressif.com/dl/package_esp32_index.json

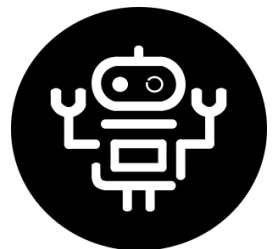
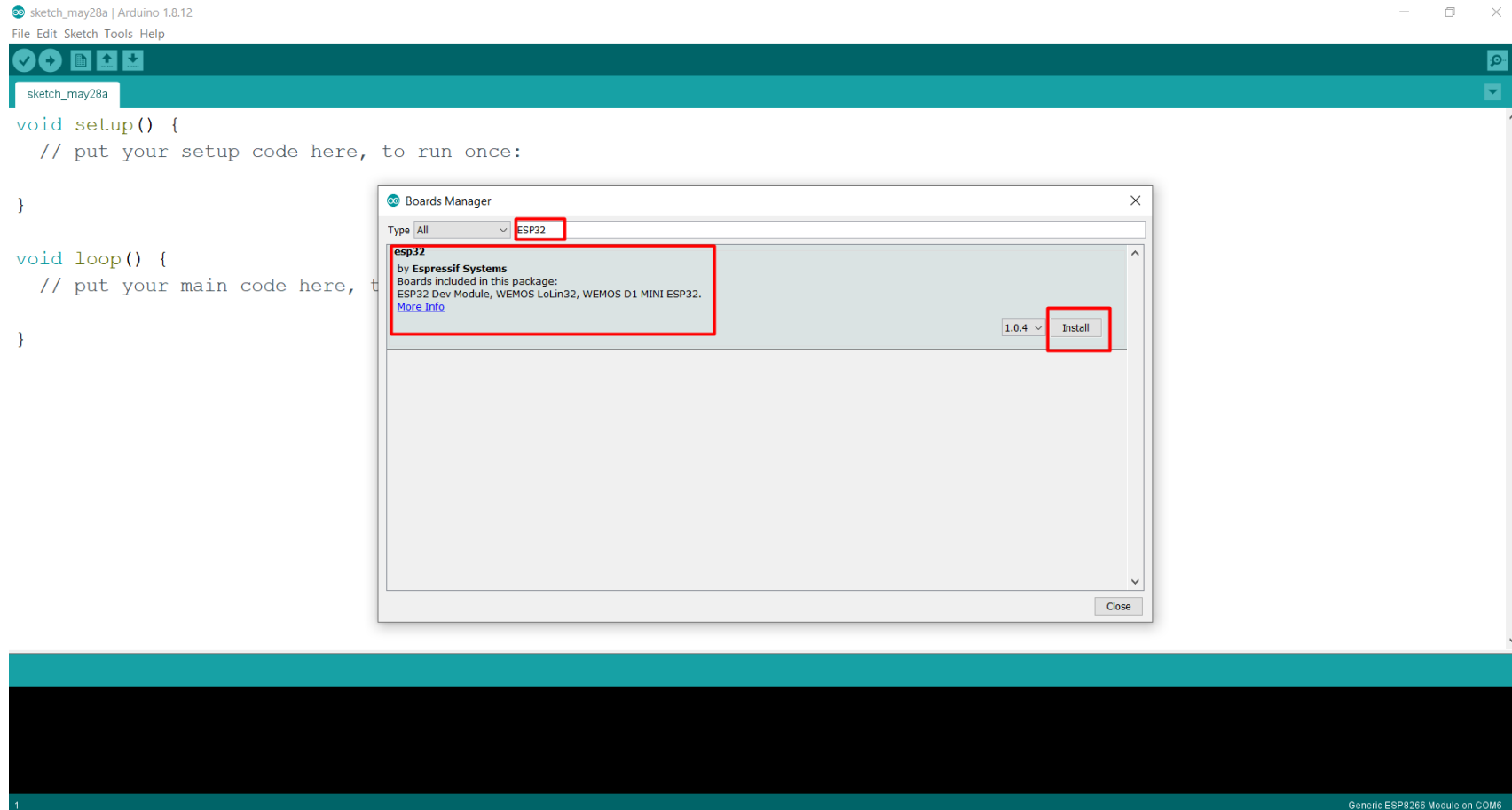


www.roboticoss.com

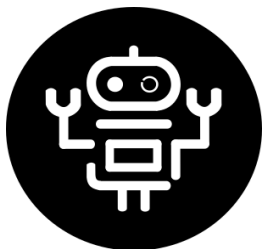
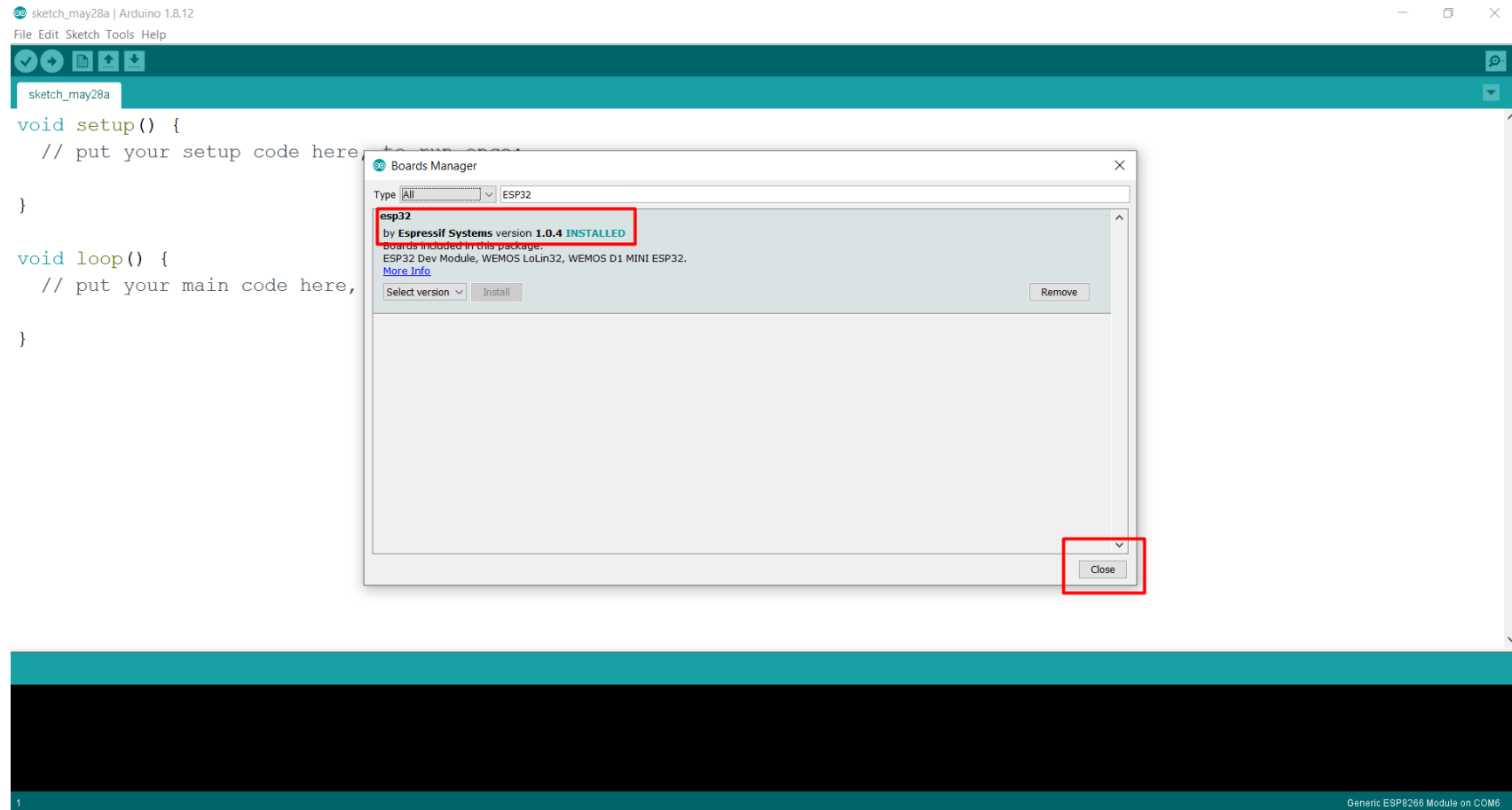
INSTALACIÓN DE LA TARJETA ESP32 EN ARDUINO IDE



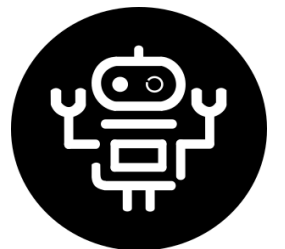
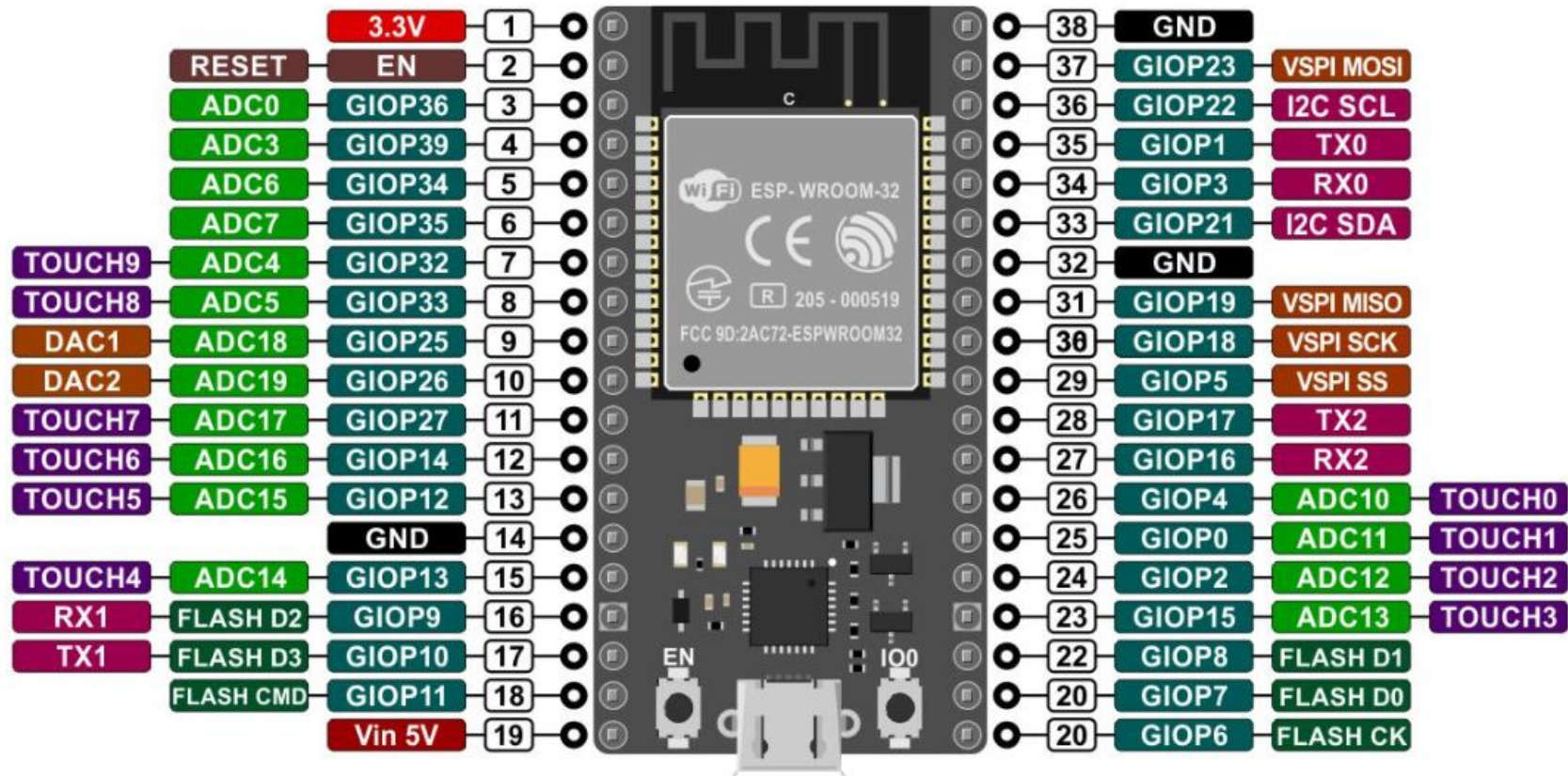
INSTALACIÓN DE LA TARJETA ESP32 EN ARDUINO IDE



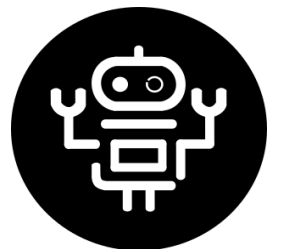
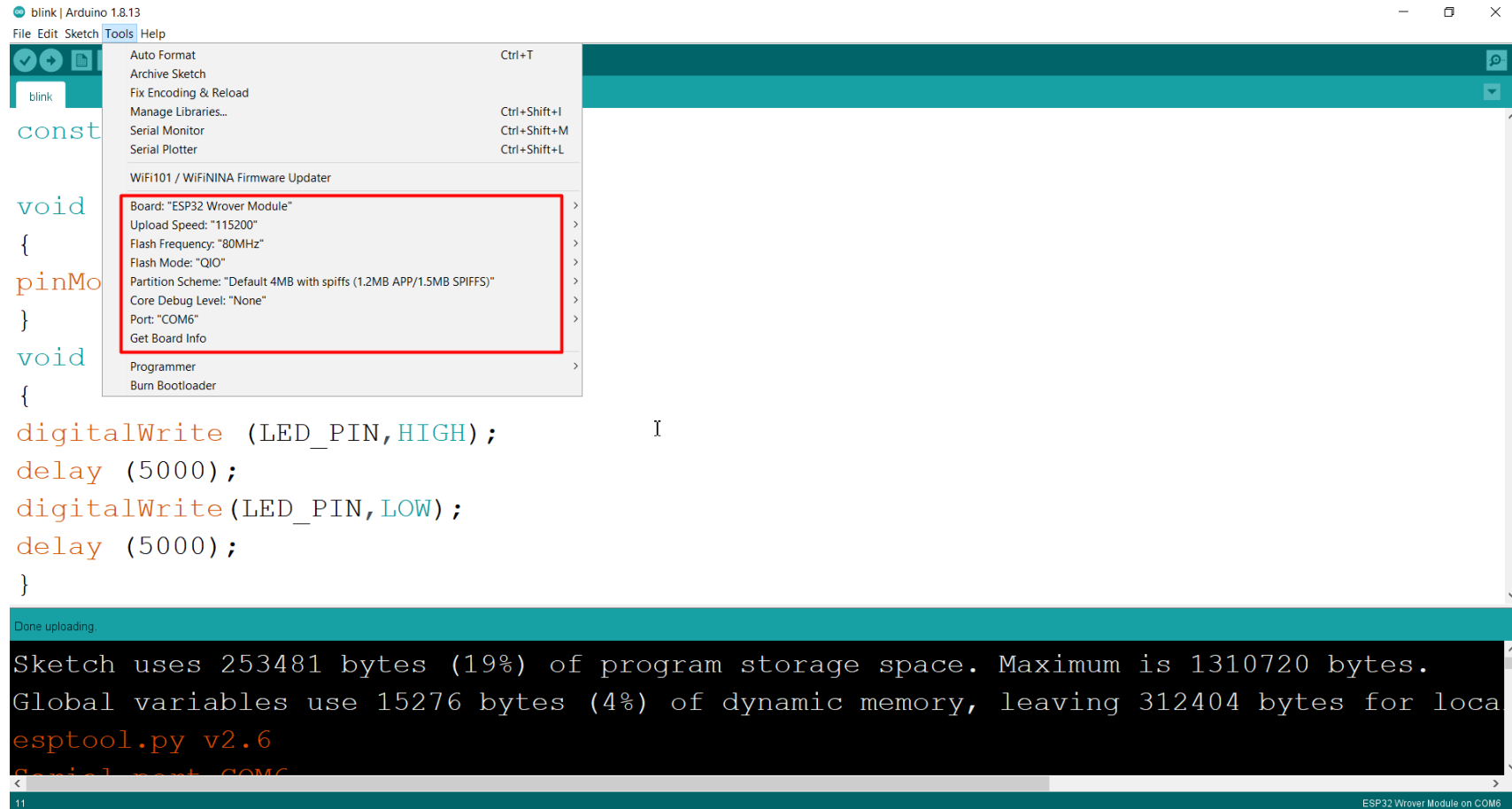
INSTALACIÓN DE LA TARJETA ESP32 EN ARDUINO IDE



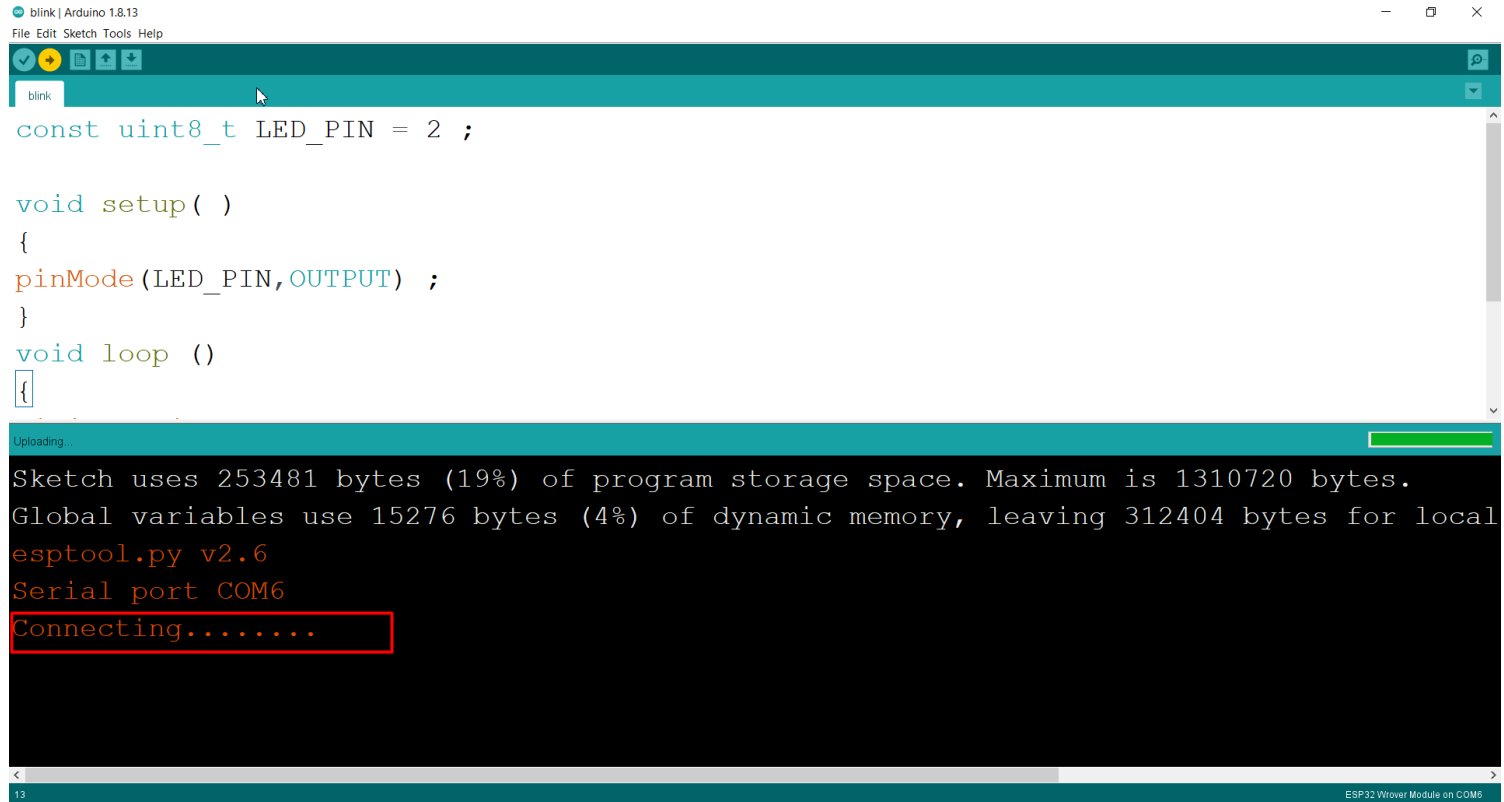
PINOUT DE LA TARJETA ESP32



CARGAR SKETCH



CARGAR SKETCH



```
blink | Arduino 1.8.13
File Edit Sketch Tools Help

const uint8_t LED_PIN = 2 ;

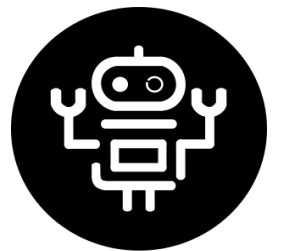
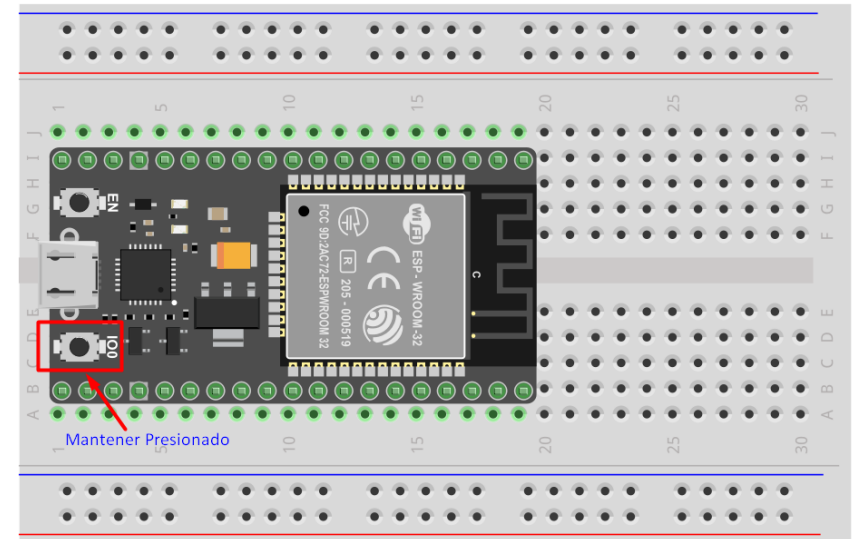
void setup( )
{
  pinMode(LED_PIN, OUTPUT) ;
}

void loop ()
{
}
```

Uploading...

Sketch uses 253481 bytes (19%) of program storage space. Maximum is 1310720 bytes.
Global variables use 15276 bytes (4%) of dynamic memory, leaving 312404 bytes for local
esptool.py v2.6
Serial port COM6
Connecting.....

13 ESP32 Wrover Module on COM6



CARGAR SKETCH

```
blink | Arduino 1.8.13
File Edit Sketch Tools Help

const uint8_t LED_PIN = 2 ;

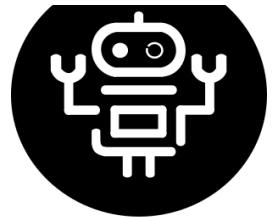
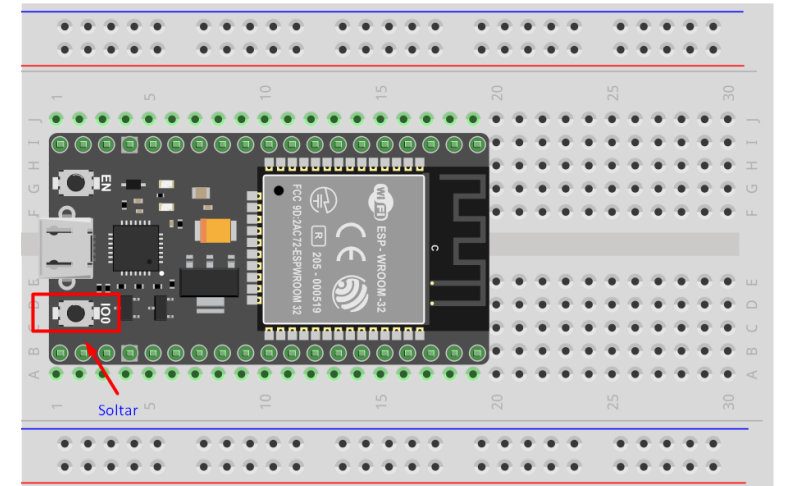
void setup( )
{
  pinMode(LED_PIN,OUTPUT) ;
}

void loop ( )
{
  // ...
}
```

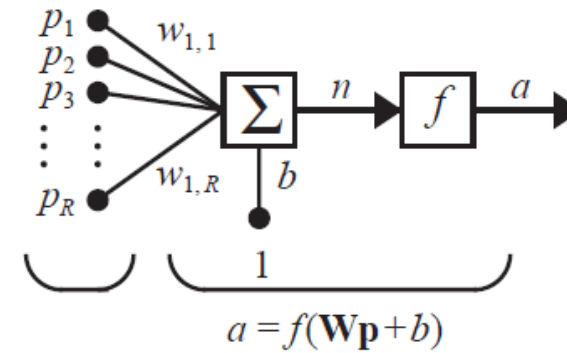
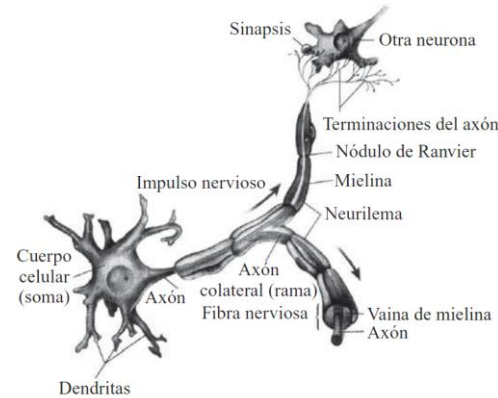
Uploading...

```
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 8192 bytes to 47...
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.0 seconds (effective 3276.8 kbit/s)
Hash of data verified.
Compressed 17392 bytes to 11186...
Writing at 0x00001000... (100 %)
```

ESP32 Wrover Module on COM6



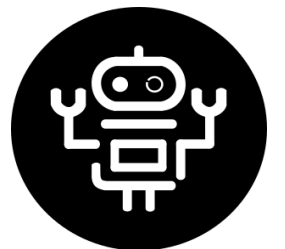
MODELO SIMPLIFICADO DE UNA NEURONA BIOLÓGICA



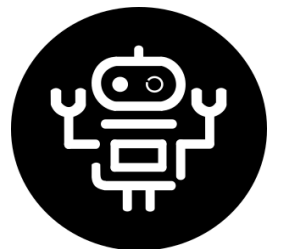
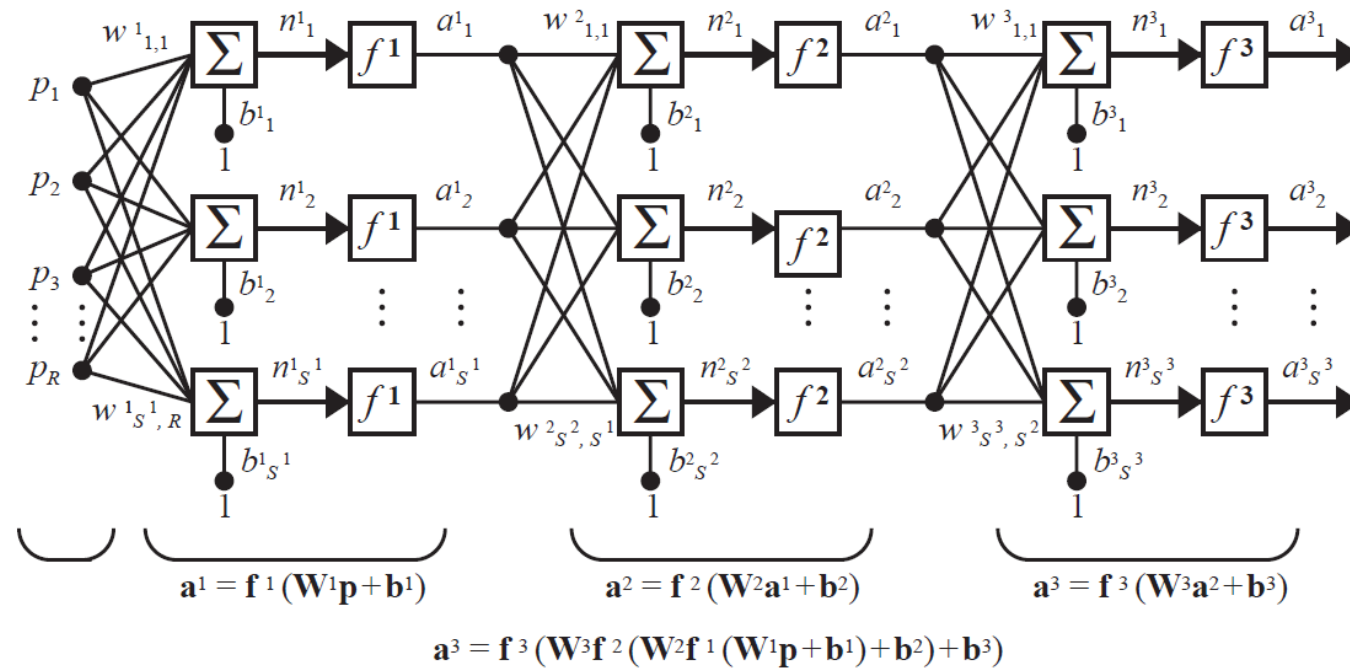
La suma neta: $n = w_{11}p_1 + w_{12}p_2 + \dots + w_{1R}p_R + b$

Forma matricial: $n = \mathbf{Wp} + b$

La salida de la neurona se calcula como: $a = f(\mathbf{Wp} + b)$



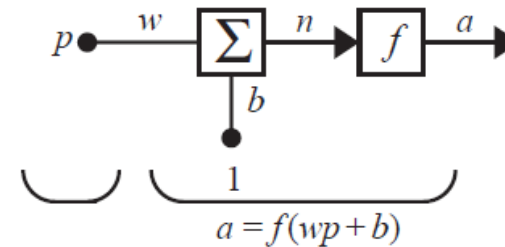
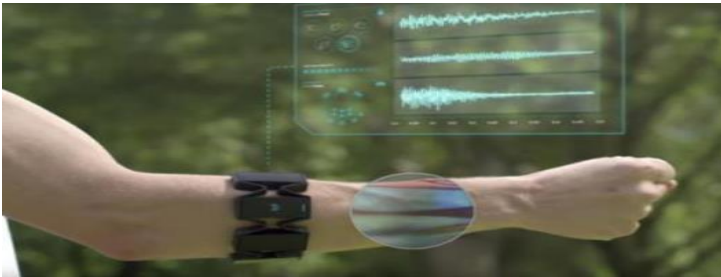
RED NEURONAL ARTIFICIAL MÚLTICAPA



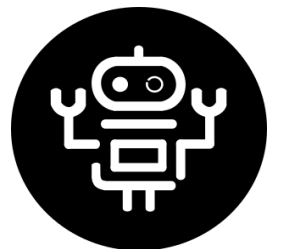
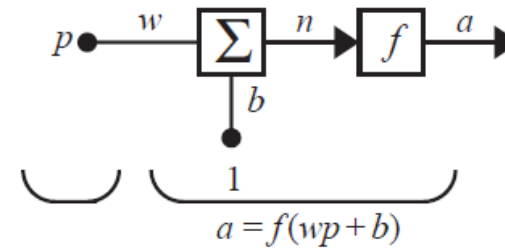
COMPONENTES DE UN RED NEURONAL ARTIFICIAL

La red neuronal artificial se compone por:

1. **Entradas (p):** Es la información que recibe la red.

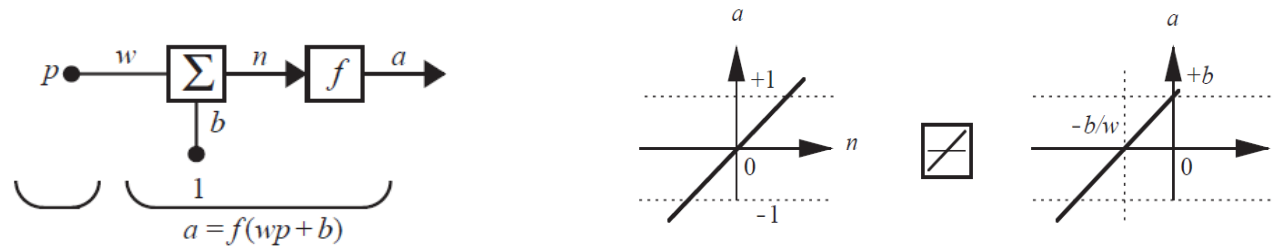


2. **Pesos (w) :** Son valores numéricos que se encargan de establecer la influencia de una entrada en la salida deseada.

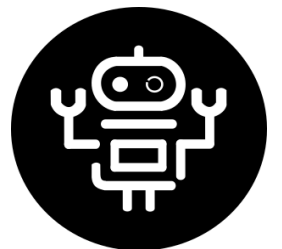
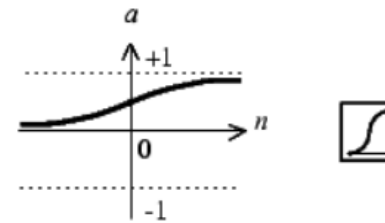
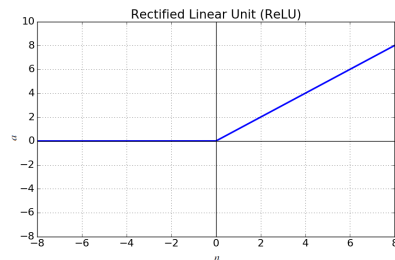


COMPONENTES DE UN RED NEURONAL ARTIFICIAL

3. Bías (b): Es muy parecido a un peso, excepto a que esta asociado a una entrada constante 1. El modelo se volverá más flexible.



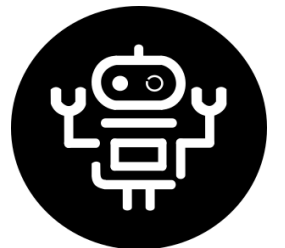
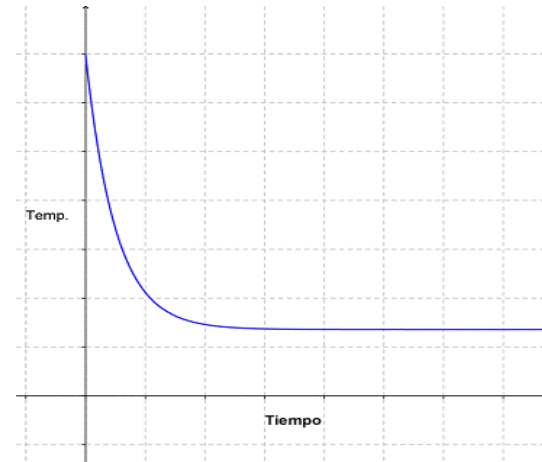
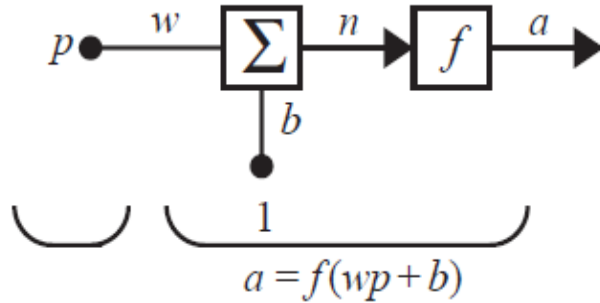
4. Función de activación o transferencia (f) : Las funciones de activación se utilizan para introducir la no linealidad en las redes neuronales.



NO LINEALIDAD EN REDES NEURONALES (FUNCIÓN DE ACTIVACIÓN)

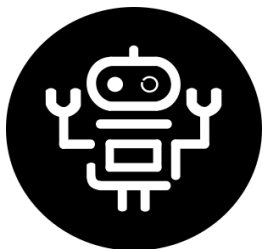
Ejemplo: Temperatura en una galleta recién horneada.

Cuando la galleta recién sale del horno la temperatura es muy alta, por lo tanto, no la puedes tocar o te quemarás, sin embargo, no toma mucho tiempo para que se enfríe lo suficiente para que la puedas comer, luego se mantienen templadas por mucho tiempo y no se enfrían más que la temperatura ambiente. Si dibujamos una gráfica del enfriamiento de una galleta, se vería más o menos así.

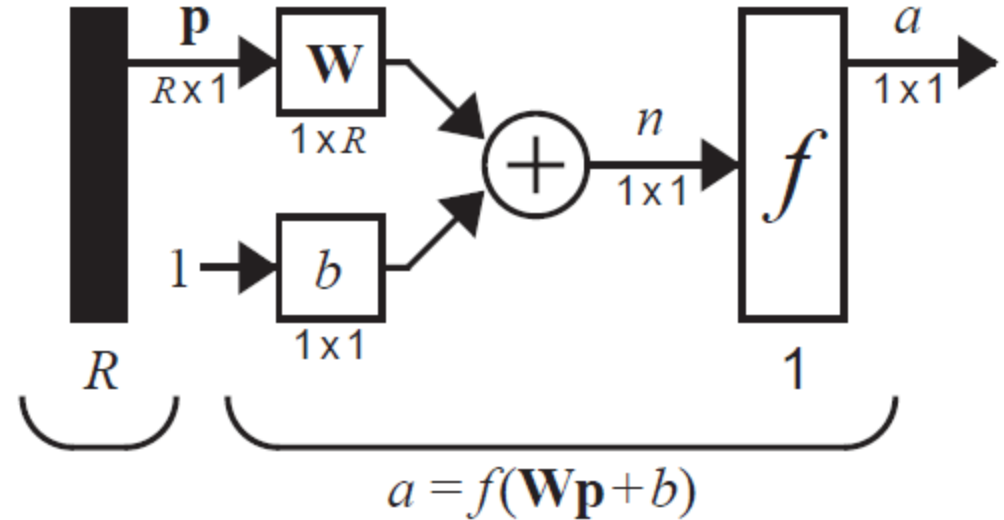
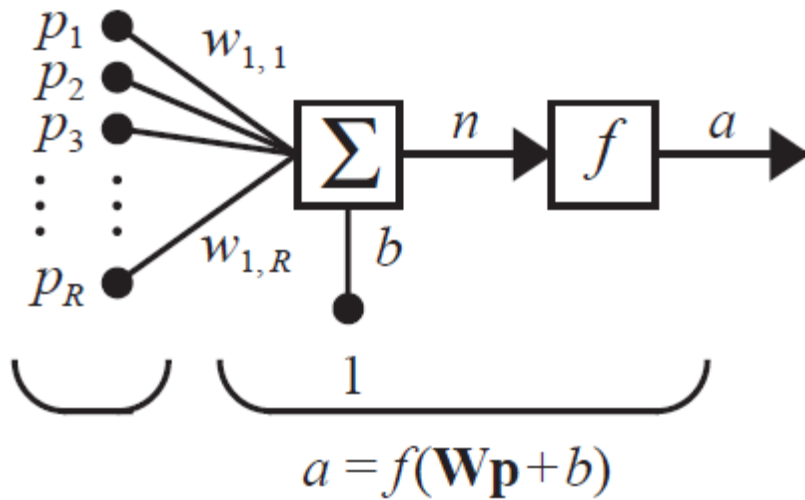


FASES DE UNA RED NEURONAL

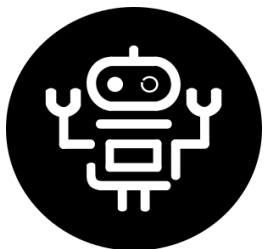
- Entrenamiento
 - Es el proceso de aprendizaje en una red neurona (Mayor costo computacional)
 - Validación
 - Es el proceso de comprobar si la red neuronal puede resolver nuevos problemas para los que ha sido entrenada.
- Predicción o inferencia.
 - Luego de entrenar y validar la red neural, ésta se implementa para que ejecute la inferencia, es decir, predecir nuevas entradas.



ARQUITECTURA DE UNA RED NEURONAL DE ENTRADAS MÚLTIPLES



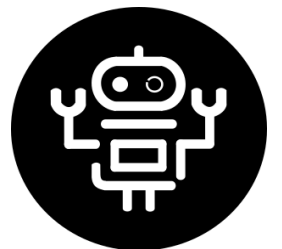
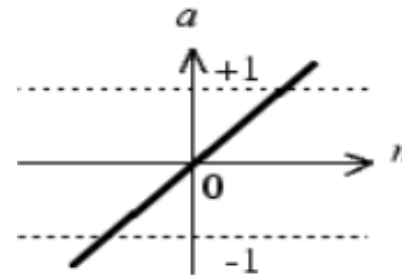
$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b.$$



FUNCIÓN DE ACTIVACIÓN LINEAL

- La función de activación es la función identidad, es decir, la salida es igual a su entrada.
- Esta función se utiliza para aproximadores de funciones.
- La salida de la neurona es continua.

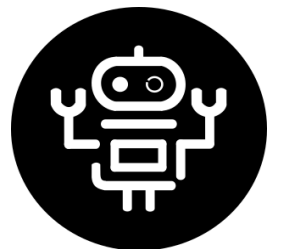
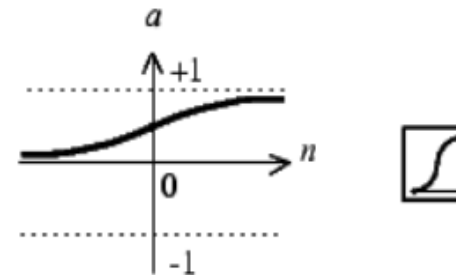
$$a = n$$



FUNCIÓN DE ACTIVACIÓN SIGMOIDAL

- Esta función de activación toma la entrada (que puede tener cualquier valor entre más y menos infinito) y reduce la salida en el rango de 0 a 1.
- Se usa comúnmente en redes multicapa que se entrenan con el algoritmo de retropropagación y en la capa final de un clasificador binario.
- La salida de la neurona es simétrica.

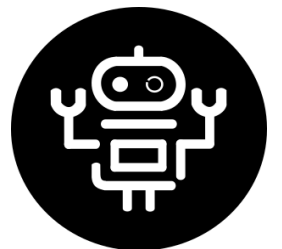
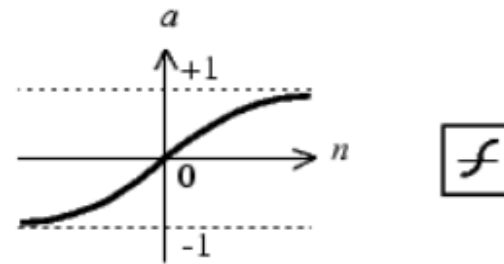
$$a = \frac{1}{1 + e^{-n}}$$



FUNCIÓN DE ACTIVACIÓN TANGENTE HIPERBÓLICA

- Esta función de activación, no es más que una versión bipolar de la función sigmoideal, según algunos autores, funciona mejor que la función logística al parecerse más a la función lineal para valores de x cercanos a cero, es decir, puede facilitar el entrenamiento de la red neuronal.
- Se usa comúnmente en redes multicapa que se entrenan con el algoritmo de retropropagación y en la capa final de un clasificador binario.

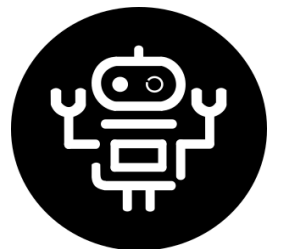
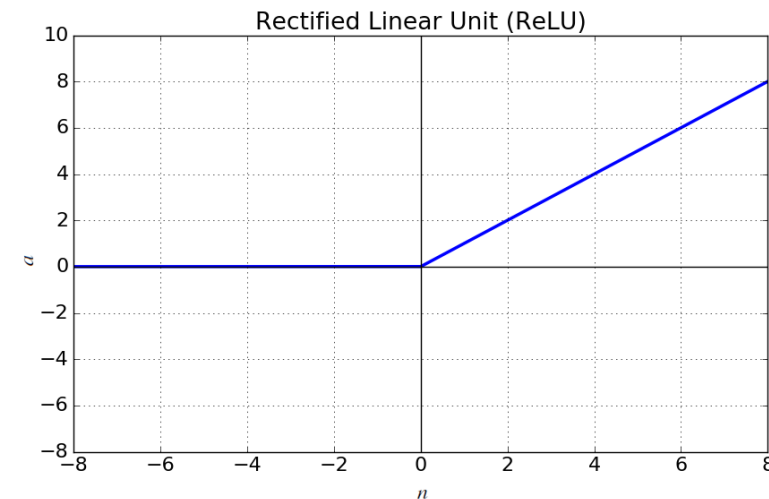
$$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$$



FUNCIÓN DE ACTIVACIÓN RELU

- Esta función de activación es extremadamente eficiente, no requiere de operaciones de tipo aritmético, su derivada es muy simple.
- Se usa comúnmente en redes multicapa que se entrenan con el algoritmo de retropropagación.
- La salida de la neurona es rectificada continua .

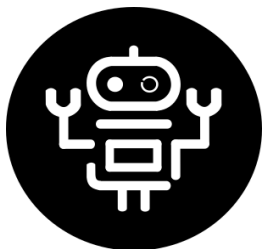
$$a = \max(0, n) = \begin{cases} 0 & n < 0 \\ n & n \geq 0 \end{cases}$$



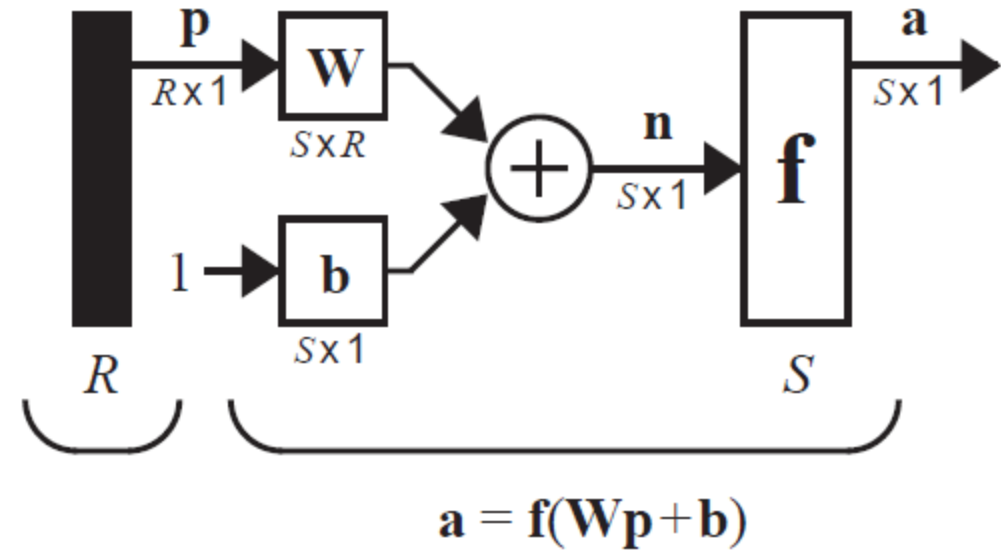
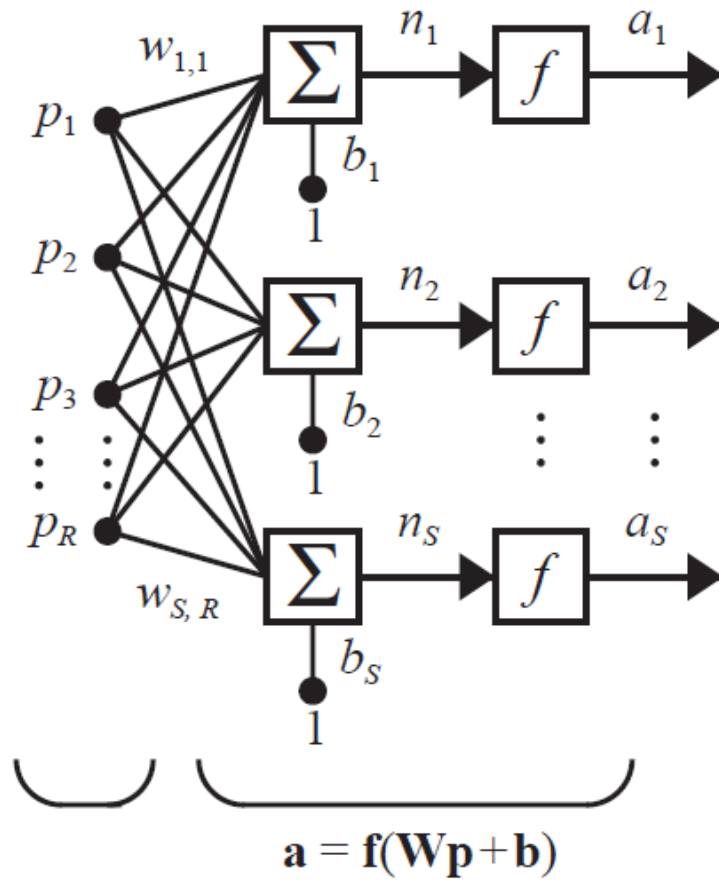
FUNCIÓN DE ACTIVACIÓN SOFTMAX

- La función Softmax transforma las salidas a una representación en forma de probabilidades, de tal manera que el sumatorio de todas las probabilidades de las salidas de 1.
- Se usa en la capa final de redes multicapa para clasificadores de múltiples clases.
- La salida de la neurona es la probabilidad de que una entrada pertenezca a una clase.

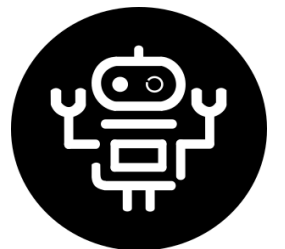
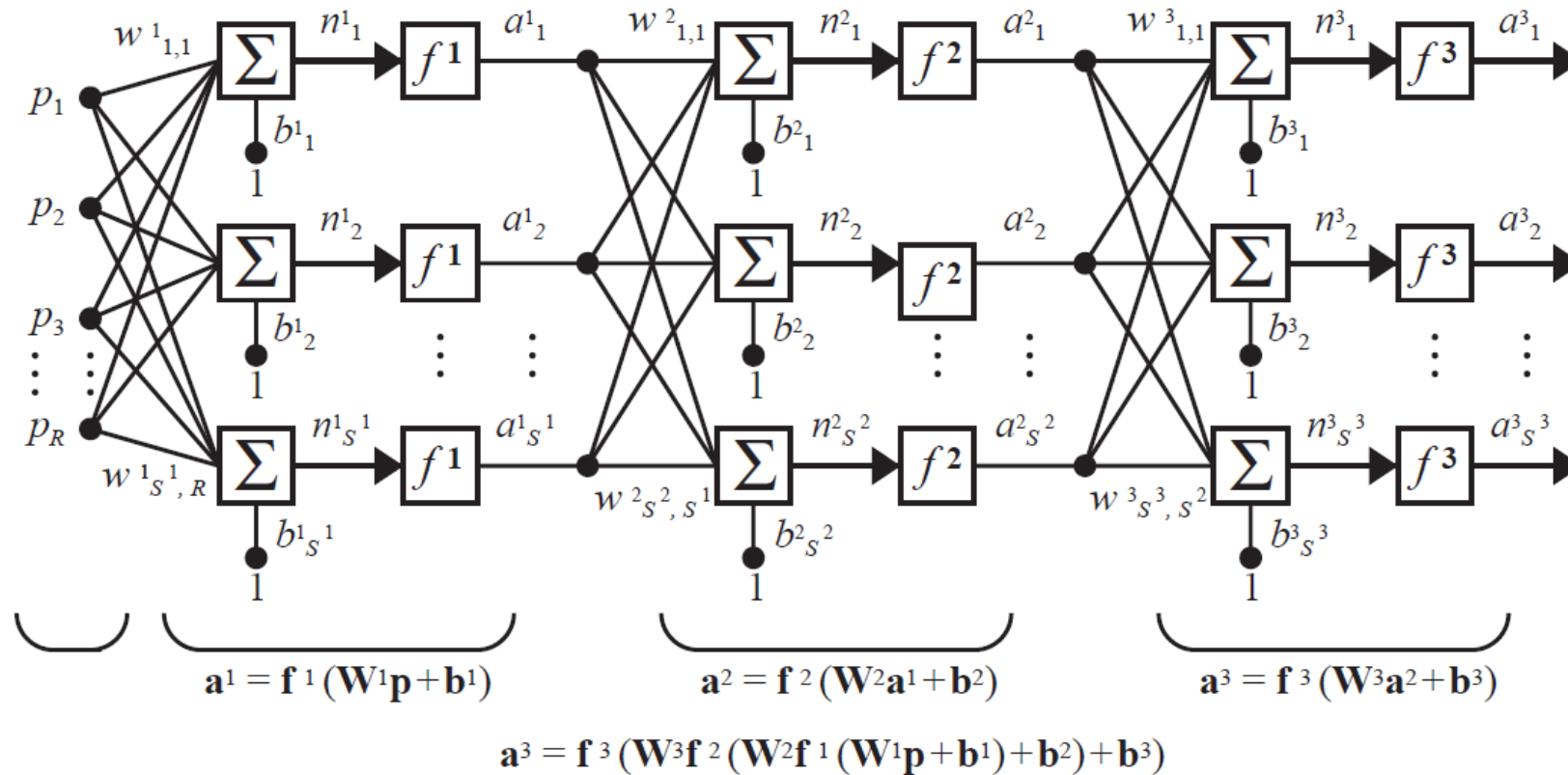
$$a_s = \frac{e^{n_s}}{\sum_{s=1}^S e^{n_s}}$$



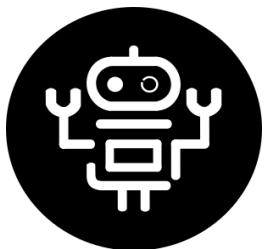
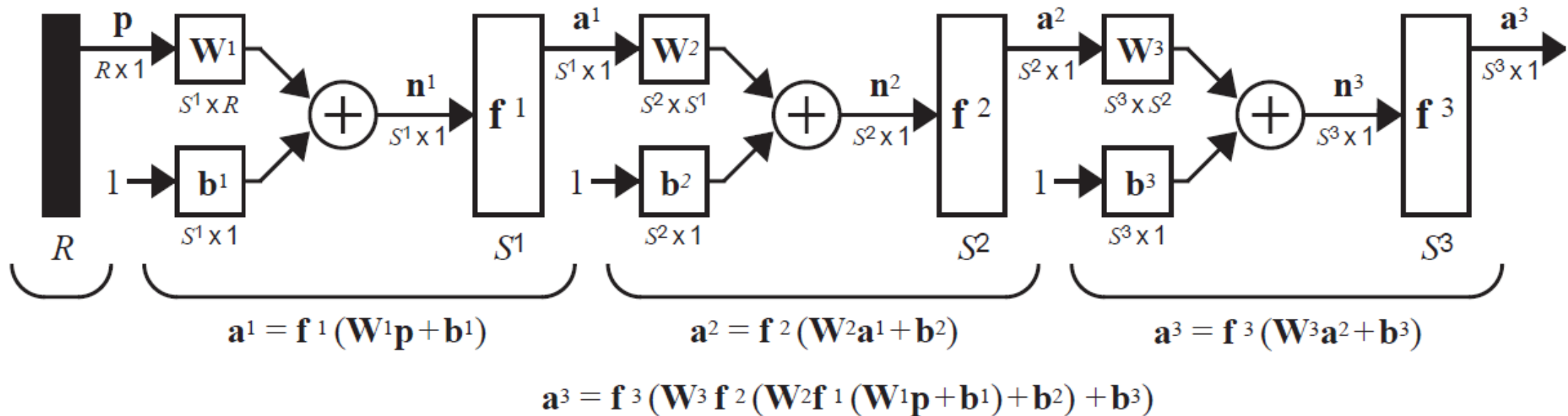
LA ARQUITECTURA DE UNA RED NEURONAL DE MÚLTIPLES ENTRADAS Y NEURONAS



LA ARQUITECTURA DE UNA RED NEURONAL DE MÚLTIPLES CAPAS DE NEURONAS

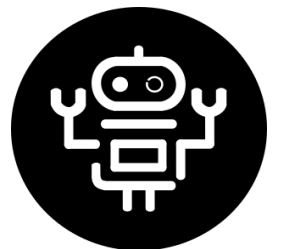


LA ARQUITECTURA DE UNA RED NEURONAL DE MÚLTIPLES CAPAS DE NEURONAS



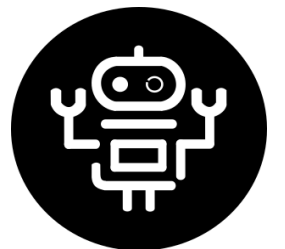
PASOS PARA EL ENTRENAMIENTO DE UNA RED NEURONAL ARTIFICIAL

- Importar o generar conjuntos de datos
- Preprocesamiento de datos.
- Dividir el conjuntos de datos en conjuntos de entrenamiento y prueba.
- Establecer hiperparámetros de algoritmo (tasa de aprendizaje, épocas).
- Definir la arquitectura de la red neuronal.
- Inicializar variables. (Pesos y bias).
- Declara las funciones de pérdida y optimizador.



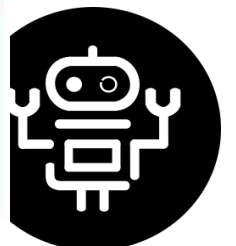
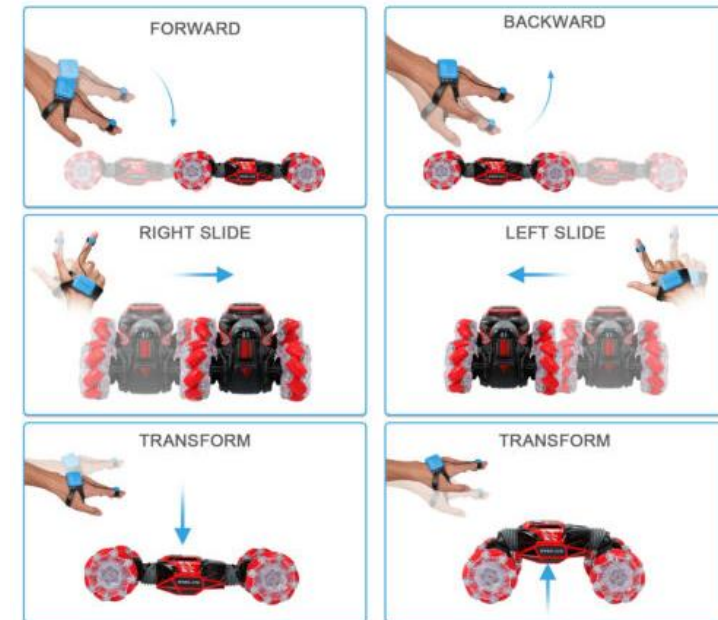
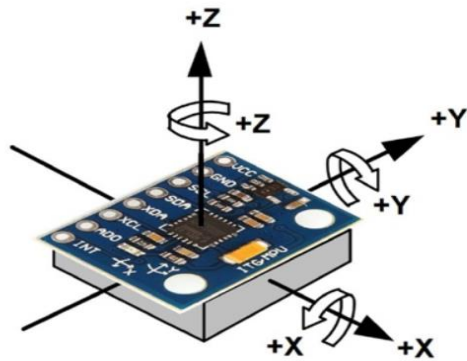
IMPORTAR - GENERAR CONJUNTOS DE DATOS

- Todos nuestros algoritmos de aprendizaje automático dependerán de los conjuntos de datos.



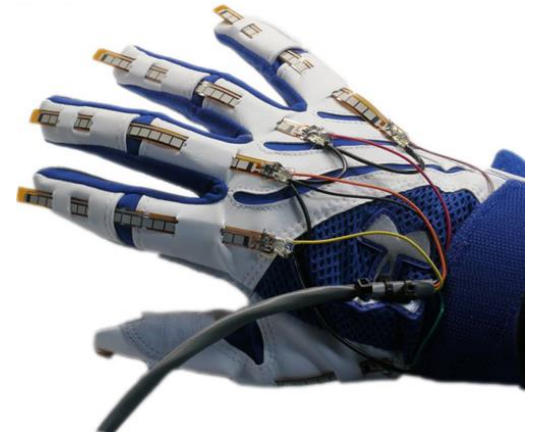
SENSOR MPU6050

- Este módulo está basado en el sensor MPU6050 y contiene todo lo necesario medir movimiento en 6 grados de libertad, combinando un giroscopio de 3 ejes y un acelerómetro de 3 ejes en un mismo chip.
- Funcionamiento:
 - Para manejar módulos con sistemas MPU6050, usaremos la biblioteca Adafruit MPU6050.



SENSOR MPU6050

- Sensor flex de 2.2", su resistividad cambia al ser flexionado. Ideal para medir el ángulo de giro en articulaciones para aplicaciones robóticas, biométricas o de interfaz de usuario.



SENSOR MPU6050

- Abra su IDE de Arduino y vaya a **Programa > Incluir biblioteca > Administrar bibliotecas** . Debería abrirse el Administrador de la biblioteca.
- Escriba " adafruit mpu6050 " en el cuadro de búsqueda e instale la biblioteca.

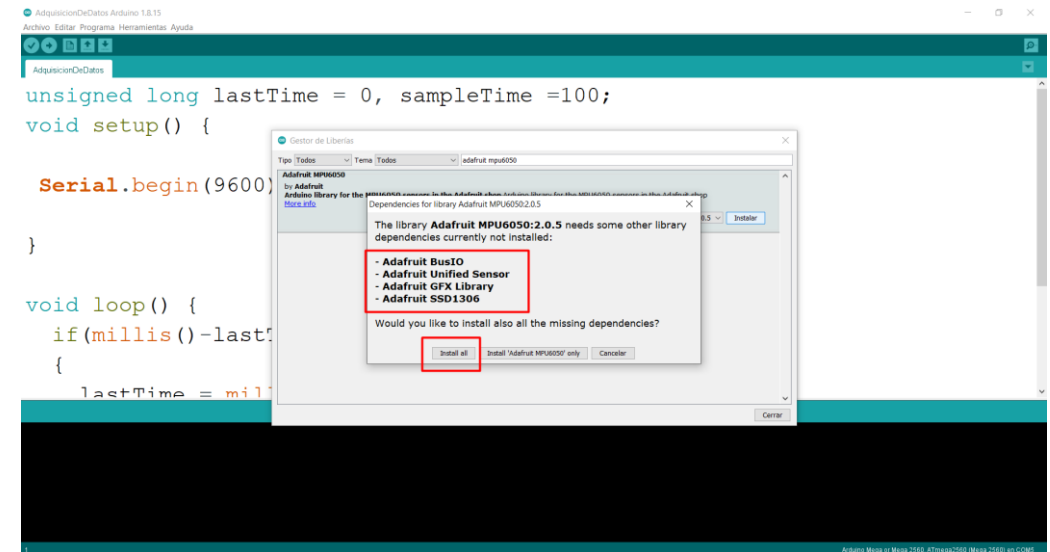
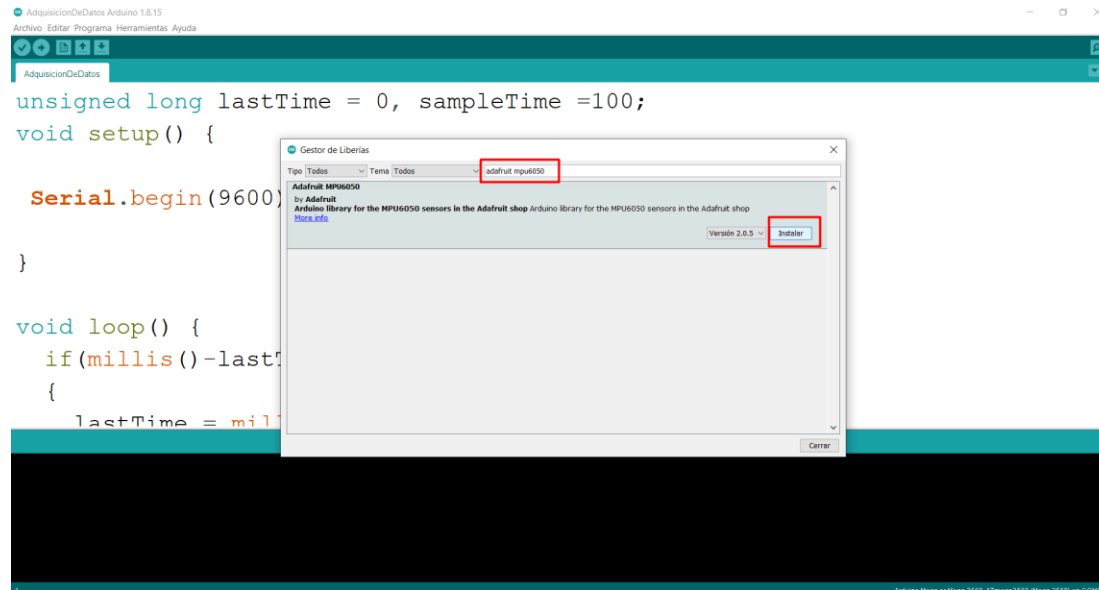
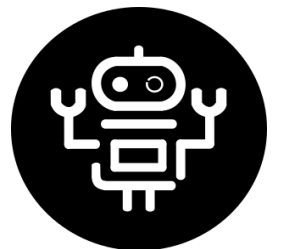
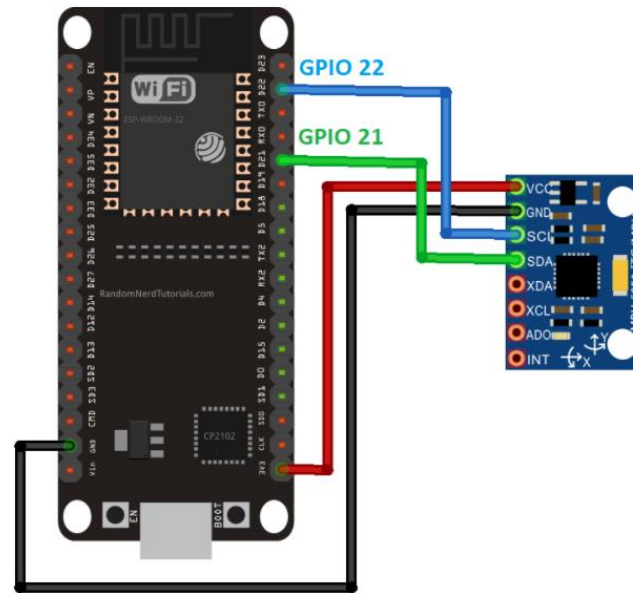
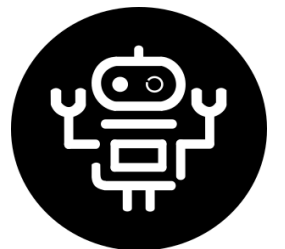
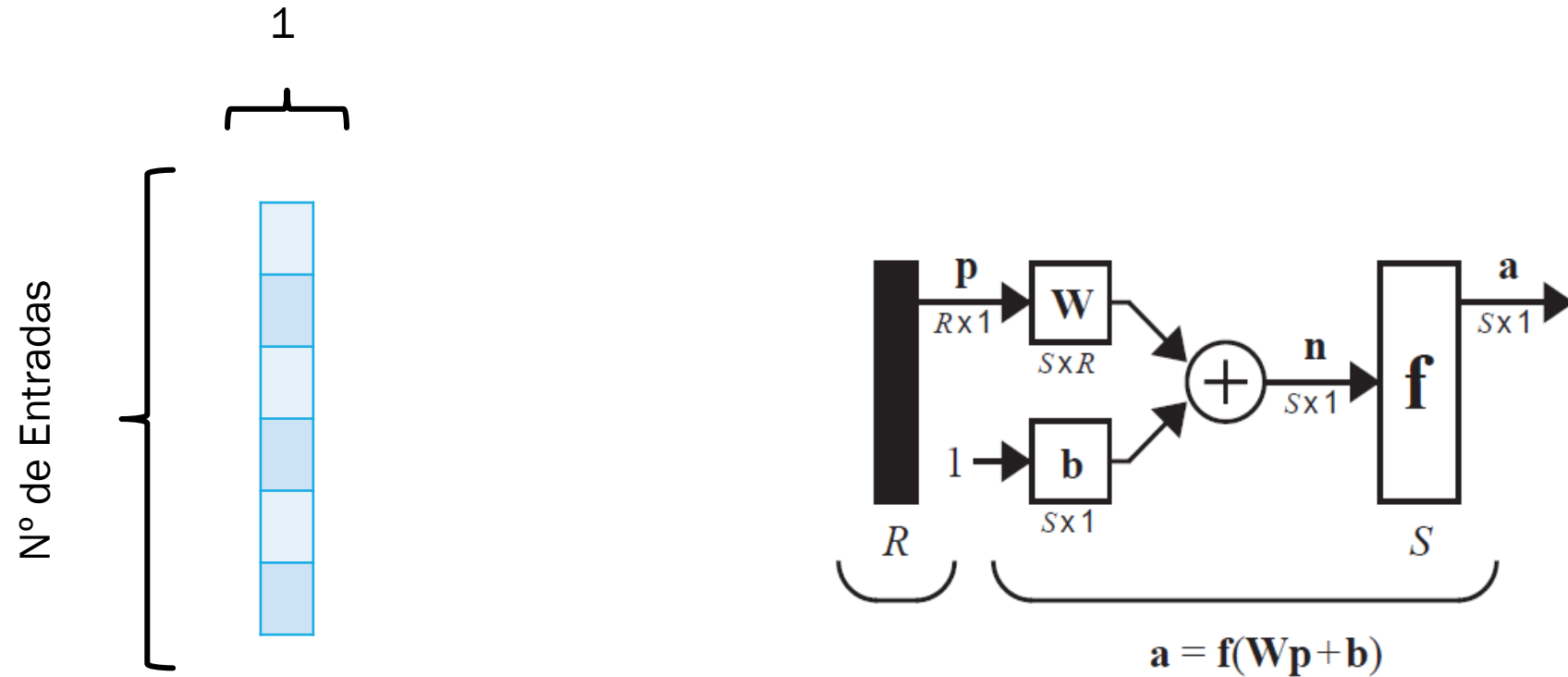


DIAGRAMA DE CONEXIONES

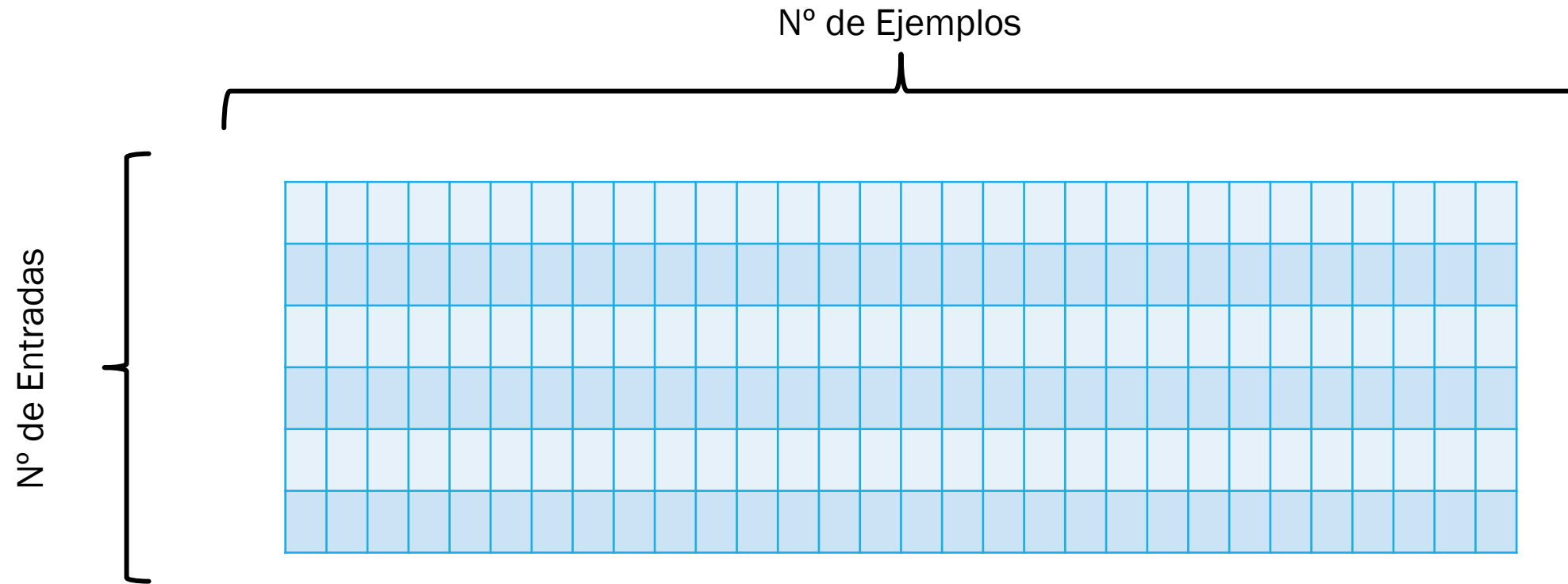
- Conecte el ESP32 al sensor MPU-6050 como se muestra en el siguiente diagrama esquemático: conecte el pin SCL a GPIO 22 y el pin SDA para GPIO 21.



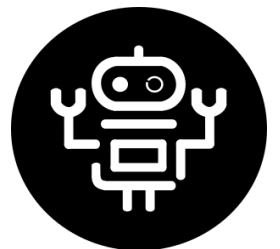
DIMENSIÓN DE DATOS DE ENTRADA EN PREDICCIÓN O INFERENCIA



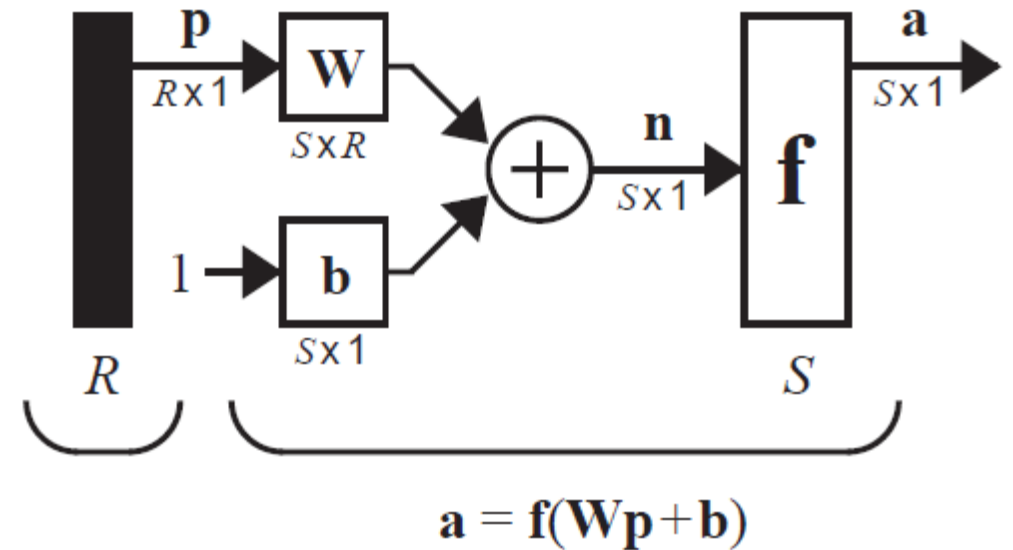
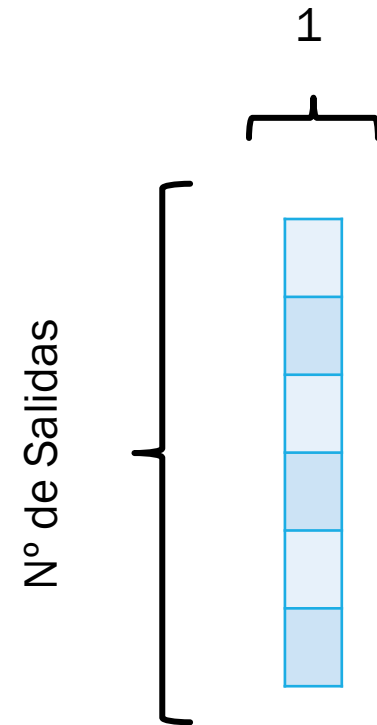
DIMENSIÓN DE DATOS DE ENTRADA EN ENTRENAMIENTO



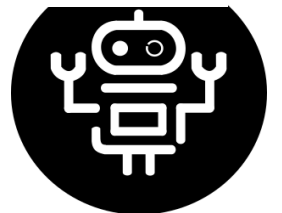
$$\mathbf{p}_{R \times Q} \xrightarrow{\text{Tensorflow}} \mathbf{p}_{Q \times R}$$



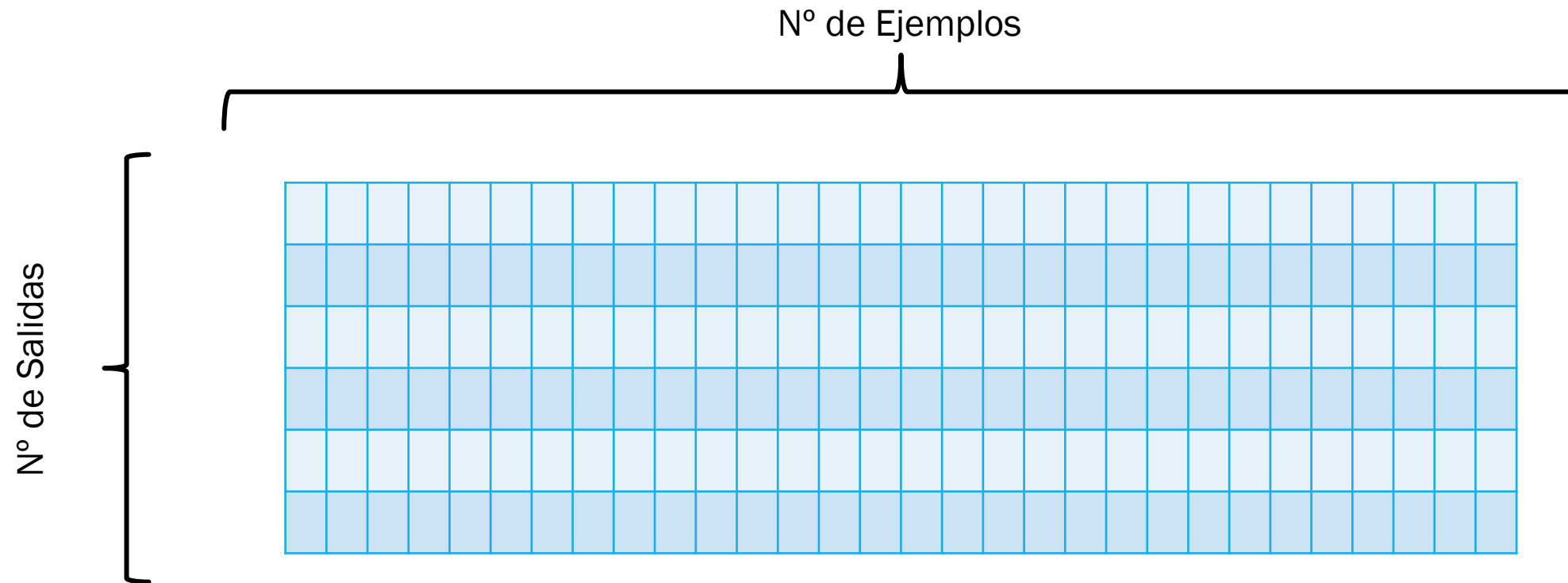
DIMENSIÓN DE DATOS DE SALIDA EN PREDICCIÓN O INFERENCIA



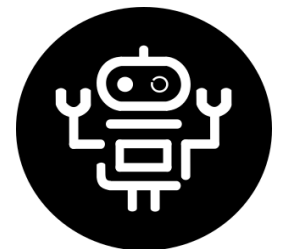
$$\mathbf{a}_{S \times 1} \xrightarrow{\text{Tensorflow}} \mathbf{a}_{1 \times S}$$



DIMENSIÓN DE DATOS DE ENTRADA EN ENTRENAMIENTO



$$a_{S \times Q} \xrightarrow{\text{Tensorflow}} a_{Q \times S}$$



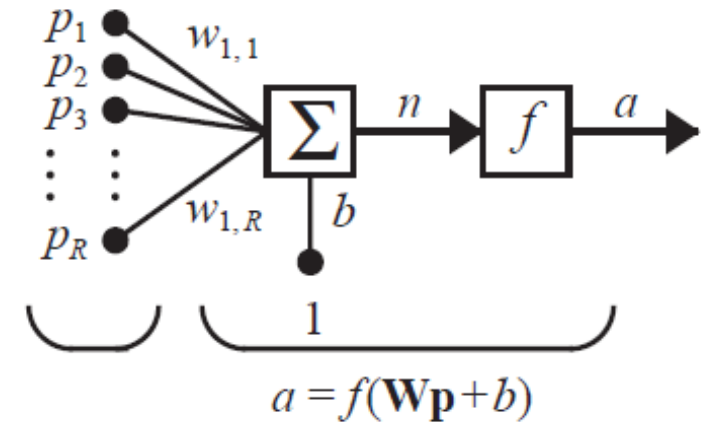
PREPROCESAMIENTO DE DATOS

- El objetivo principal de la etapa de preprocesamiento de datos es facilitar la capacitación de la red.
- El preprocesamiento de datos consta de pasos tales como:
 - Normalización
 - Transformaciones no lineales
 - Extracción de características
 - Codificación de entradas y salidas
 - Manejo de datos faltantes, etc.



$$p^n = 2(p - p^{min}) ./ (p^{max} - p^{min}) - 1$$

$$p^n = (p - p^{mean}) ./ p^{std}$$



NORMALIZACIÓN

- Se recomienda normalizar los datos que usan escalas e intervalos diferentes.
- El modelo con datos sin normalización podría adaptarse automáticamente a datos tan heterogéneos, pero pueden ralentizar o dificultar el aprendizaje.
- Aquí vamos a usar StandardScaler de Scikit-learn
- Para cada característica reste la media y escale los datos con desviación estándar 1.
- <https://scikit-learn.org/stable/modules/preprocessing.html>

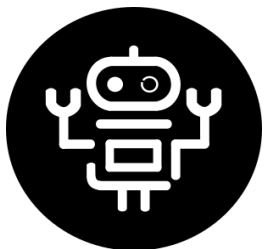
$$\mathbf{p}^n = (\mathbf{p} - \mathbf{p}^{mean}) ./ \mathbf{p}^{std}$$



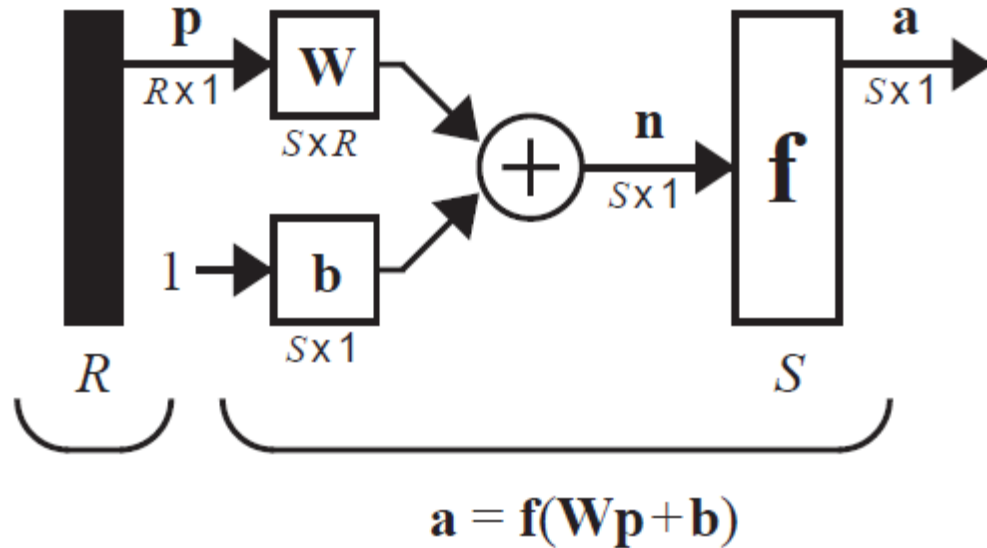
INSTALACIÓN SCIKIT-LEARN

```
Command Prompt - pip install -U scikit-learn
Microsoft Windows [Version 10.0.18362.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Edison>pip install -U scikit-learn
Collecting scikit-learn
  Downloading https://files.pythonhosted.org/packages/92/db/8c50996186faed765392cb5ba495e8764643b71adbd168535baf0fcae5f1
/scikit_learn-0.23.2-cp37-cp37m-win_amd64.whl (6.8MB)
    | 286kB 152kB/s eta 0:00:43
```

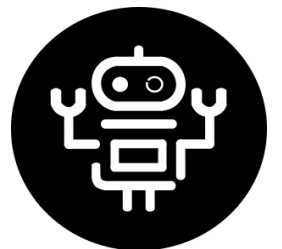


CODIFICACIÓN DE SALIDAS



$a[0]$	$a[1]$	Etiqueta
0	0	adelante
0	1	derecha
1	0	izquierda
1	1	atrás

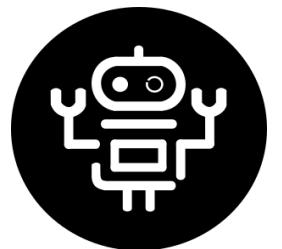
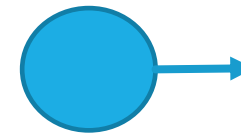
Si tenemos n salidas en la capa final de la red neuronal, obtenemos 2^n clases.



CODIFICACIÓN BINARIA

- Una etiqueta toma el valor de 1 y la otra 0.
- Se utiliza para problemas de clasificación binaria.

Movimiento (Texto)	Codificación (Numérico)
derecha	0
izquierda	1

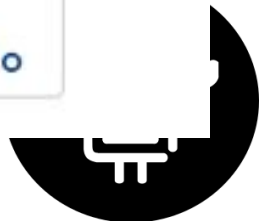


CODIFICACIÓN MEDIANTE ENTEROS

- A cada etiqueta se le asigna un valor entero.
- Es fácilmente reversible y podemos obtener la etiqueta a partir del número.
- Este tipo de codificación solo es apropiado si existe una relación de orden entre las categorías, como “primero”, “segundo” y “tercero”. Este tipo de datos categóricos se conoce como ordinal.

¿Qué tan satisfecho te encuentras con nuestro servicio?

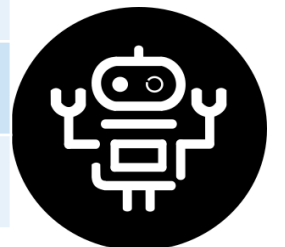
				
☐	☐	☐	☐	☐
Muy insatisfecho	Insatisfecho	Neutral	Satisfecho	Muy satisfecho



CODIFICACIÓN CATEGÓRICA ONE HOT ENCODING

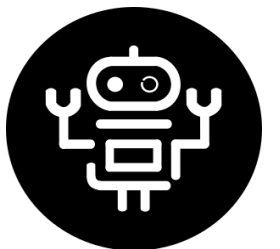
- Se llama one-hot porque solo un bit es "hot" o 1 en cualquier momento, esta técnica es comúnmente utilizada para trabajar con características categóricas (Multi-class Classification Problem).
- La idea es asignar 0 a toda las clases, excepto para la clase a la que pertenecen los datos, este caso asignamos un 1.
- Aqui vamos a utilizar `keras.utils.to_categorical` para la codificación One-Hot.
- Para implementar esta herramienta los datos deben estar codificados mediante enteros

Etiqueta	ID	One Hot Encoding		
		Rosa	Lirio	Girasol
Rosa	0	1	0	0
Lirio	1	0	1	0
Girasol	2	0	0	1



DECLARAR LA FUNCIÓN DE PÉRDIDA.

- **Funciones de pérdida de regresión**
 - Pérdida de error medio cuadrático
 - Pérdida de error medio absoluto
 - Pérdida de error cuadrático medio logarítmico
- **Funciones de pérdida de clasificación binaria**
 - Entropía cruzada binaria
- **Funciones de pérdida de clasificación de clase múltiple**
 - Entropía cruzada multiclase

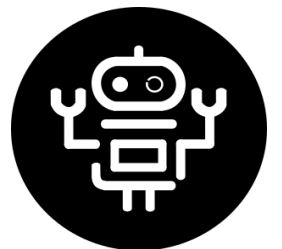
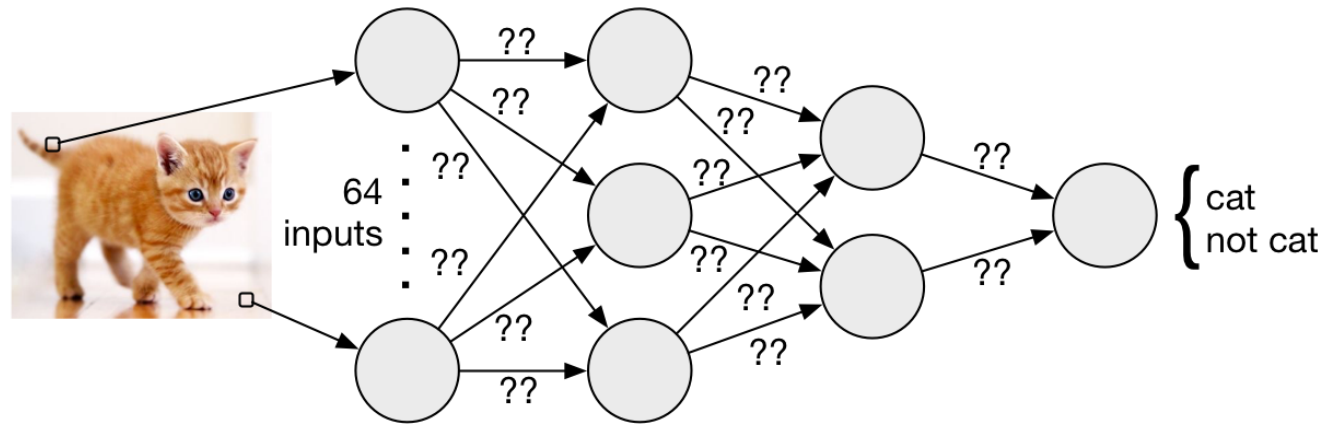


ENTROPÍA CRUZADA BINARIA

loss='binary_crossentropy'

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^N (y \times \log(\hat{y}_i) + (1 - y) \times \log(1 - \hat{y}_i))$$

- La función de pérdida de entropía cruzada binaria se utiliza en las decisiones si/no, donde el modelo intenta decidir si la o las entradas pertenecen o no a una clase.

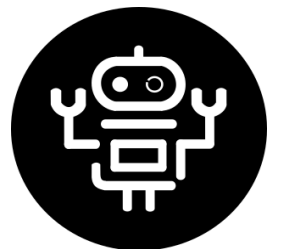
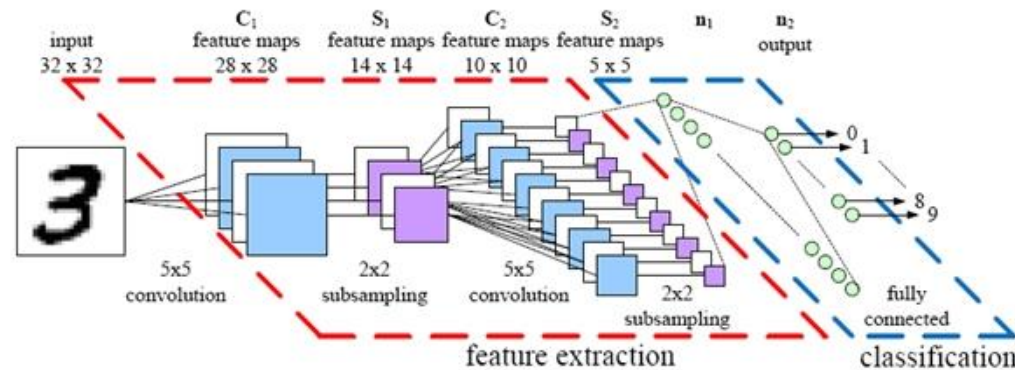


ENTROPÍA CRUZADA CATEGÓRICA

loss='categorical_crossentropy'

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{j=1}^M \sum_{i=1}^N (y_{ij} * \log(\hat{y}_{ij}))$$

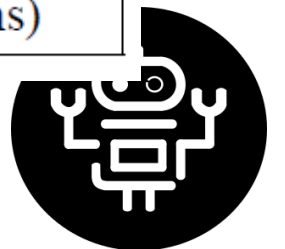
- La entropía cruzada categórica es la función de pérdida predeterminada para problemas de clasificación de varias clases.
- La función requiere que la capa de salida esté configurada con n neuronas (uno para cada clase), y una función de activación softmax para predecir la probabilidad de cada clase.



OPTIMIZADORES

- Los optimizadores son algoritmos o métodos que se utilizan para cambiar los atributos de su red neuronal, como los pesos y la tasa de aprendizaje con el objetivo para reducir la pérdida (error).
- La elección del algoritmo de optimización para su modelo de aprendizaje puede significar la diferencia entre buenos resultados en minutos u horas.

model/accuracy	training	validation	test
simple	89.96%	90.70%	90.71%
2 hidden(128)	90.81%	91.40%	91.18%
dropout(30%)	91.70%	94.42%	94.15% (200 epochs)
RMSProp	97.43%	97.62%	97.64% (10 epochs)
Adam	98.94%	97.89%	97.82% (10 epochs)

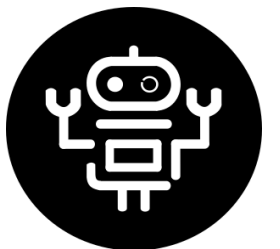


GRADIENTE DESCENDENTE ESTOCÁSTICO (SGD)

- Es el algoritmo es probablemente el más popular y ampliamente utilizado de todos los optimizadores.
- La simplicidad de SGD lo convierte en una buena opción para redes poco profundas.
- Sin embargo, el SGD converge significativamente más lentamente que otros algoritmos más avanzados.

optimizer = 'sgd'

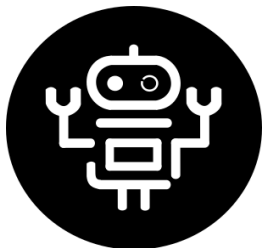
tf.keras.optimizers.SGD(learning_rate=lr)



ADAM

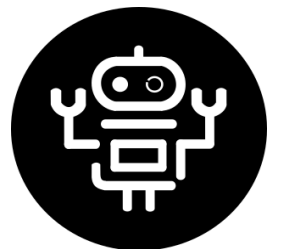
- Adam es un optimizador mejorado.
- Adam proporciona tanto una tasa de aprendizaje adaptativa como la estimación adaptativa de momentos de primer y segundo orden.
- Adam es un algoritmo popular en el campo del aprendizaje profundo porque logra buenos resultados rápidamente.
- Como recomendación es dejar los hiperparámetros de algoritmo en su configuración predeterminada.

optimizer = 'adam'



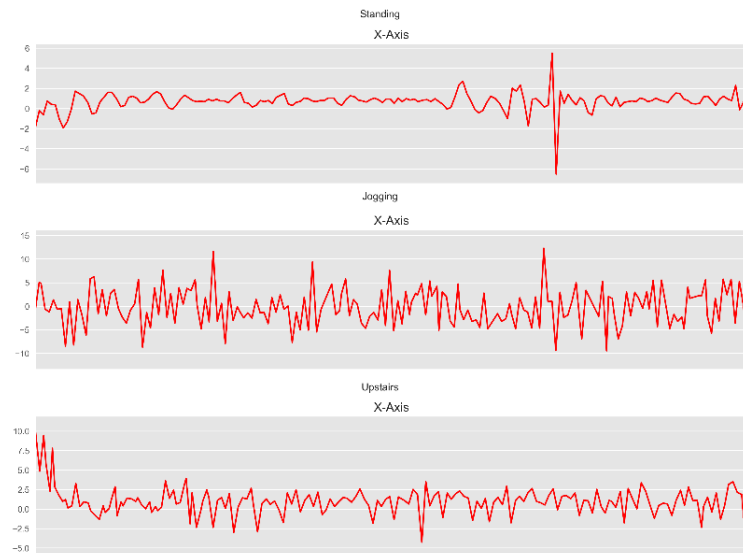
SERIES TEMPORALES

- Una serie temporal se define como una colección de observaciones de una variable recogidas secuencialmente en el tiempo.
- Estas observaciones se suelen recoger en instantes de tiempo equi-espaciados, o bien acumular los valores sobre intervalos de tiempo.



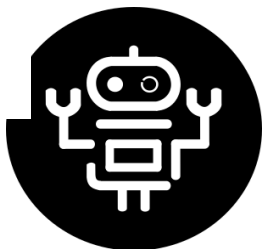
EXTRACCIÓN DE CARACTERÍSTICAS

- Esta etapa consiste en extraer la información más relevante de la sección de un patrón.
- Minimizar la complejidad de la implementación.
- Reducir el costo computacional del procesamiento de la información.



$$\overline{X_i} = \frac{1}{N} \sum_{k=1}^N |x_k|, \quad i=1, 2, \dots, I,$$

$$S_x(f) = \lim_{T \rightarrow \infty} E \left\{ \frac{1}{2T} \int_{-T}^T x(t) e^{-j2\pi ft} \right\}$$

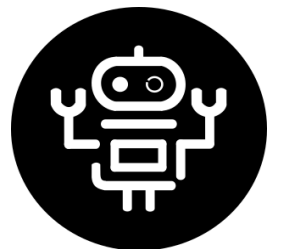


VALOR MEDIO ABSOLUTO (MAV)

- Es el promedio de los valores absolutos de las N muestras tomadas.
- El MAV mide el nivel de contracción de las señales EMG.



$$MAV = \frac{1}{N} \sum_{k=1}^N |p_k|$$

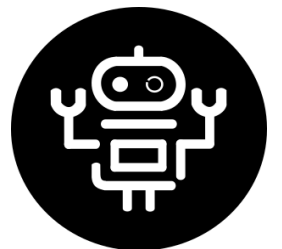


VALOR EFICAZ (RMS)

- Es un valor escalar, correspondiente a la media cuadrática.
- Es usado porque ayuda a obtener la amplitud de la señal en el movimiento muscular (EMG).



$$RMS = \sqrt{\frac{1}{N} \sum_{k=1}^N p_k^2}$$



LONGITUD DE FORMA DE ONDA (WL)

- Es una variación acumulativa que puede indicar el grado variaciones de las señales.



$$WL = \sum_{k=1}^N |p_k - p_{k-1}|$$