

## Guía para ejecutar el código

Esta es una breve guía que explica cómo instalar y configurar el sistema para poder ejecutar el código del proyecto sin problemas.

La guía está orientada para usuarios de **Windows** (en especial, **Windows 10**), aunque los pasos son muy similares para otros sistemas operativos.

### Instalación del intérprete de Python

Python es un lenguaje interpretado. Para poder ejecutar código en Python necesitamos sí o sí instalar su intérprete en nuestro sistema.

*La versión del intérprete que estamos utilizando es la 3.6.7.*

Utilizamos esta versión debido a que es la que funciona con el código que estamos desarrollando actualmente. Versiones posteriores nos causaron problemas de compatibilidad.

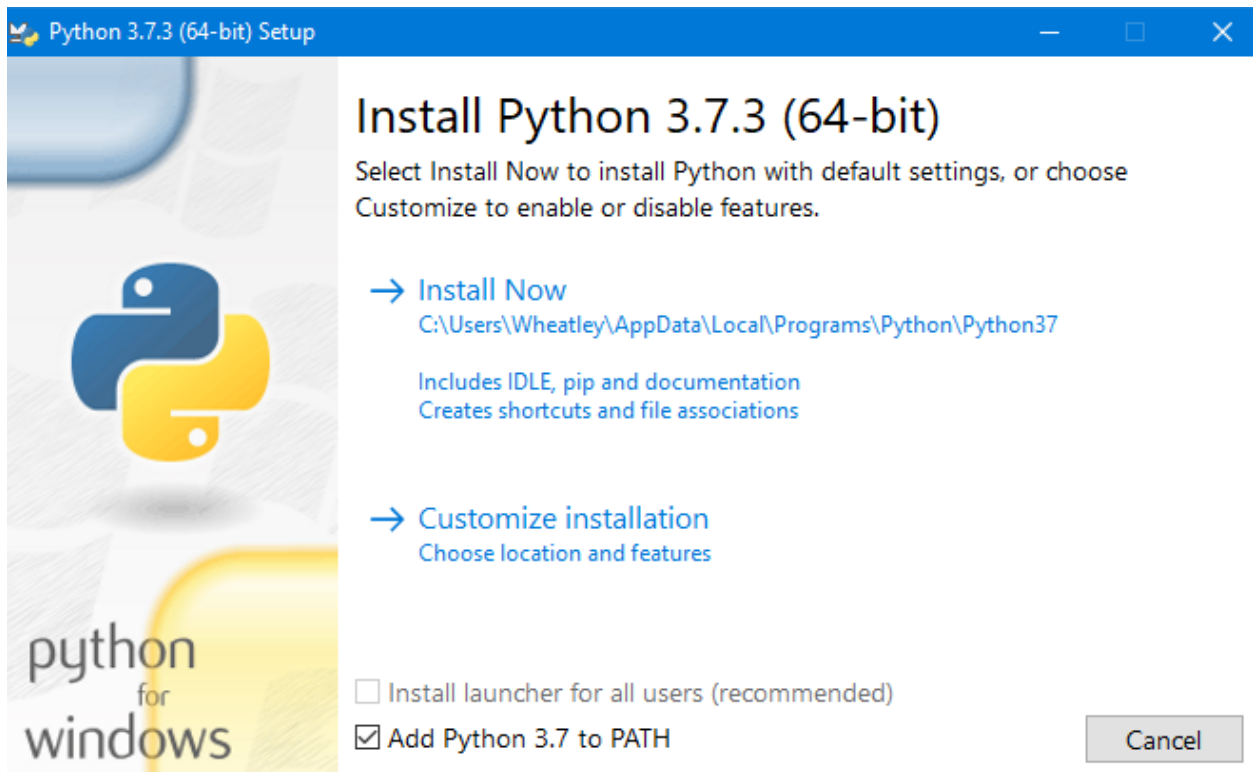
En este enlace se encuentran las descargas para los diferentes sistemas operativos:

<https://www.python.org/downloads/release/python-367/>

En este enlace está la descarga directa para Windows:

<https://www.python.org/ftp/python/3.6.7/python-3.6.7-amd64.exe>

Al abrir el ejecutable para instalarlo, tenemos que activar sí o sí la opción que **dice Add Python <versión> to PATH**:



Esta opción agrega la ruta de instalación del intérprete a nuestra variable de entorno PATH del sistema operativo. Lo que esto nos permite es (cuando abramos una consola de comandos de Windows (CMD) o algún otro tipo de prompt (puede ser Powershell también)) *el hecho de poder llamar al intérprete de Python sin tener que especificar la ruta completa de donde se encuentra.*

Por esto mismo **es altamente recomendable** aplicar esta opción.

## Uso de Pip, instalación de paquetes

Pip es un sistema de gestión de paquetes (packages en inglés) de Python. Su nombre hace referencia al Python Package Index, un índice donde se encuentran la mayoría de los paquetes de Python. Esta herramienta nos simplifica el hecho de instalar, actualizar, o remover paquetes que utilizaremos en nuestros códigos.

*Los paquetes son similares a las librerías que se utilizan en otros lenguajes de programación de alto nivel. Son un conjunto de herramientas, referencias a otros paquetes, módulos, etc.*

Para instalar un paquete con Pip, simplemente tenemos que abrir una interfaz de comandos (puede ser el símbolo de sistema de Windows (CMD), o Powershell) y escribir:

***pip install <paquete>***

Esto nos instalará la última versión estable y disponible del paquete indicado. Si queremos instalar una versión específica, debemos hacer lo siguiente:

***pip install tensorflow==1.14***

Donde, luego de los símbolos ==, indicamos el número de versión.

## Paquetes y requisitos necesarios

### TensorFlow

TensorFlow es el paquete que nos simplifica la tarea de crear y entrenar una red neuronal. Este es el paquete central de nuestro proyecto.

Debido a problemas de compatibilidad con el código de nuestro proyecto, la versión de TensorFlow que estamos utilizando no es la última de todas.

### El código funciona con las siguientes versiones de TensorFlow:

- **tensorflow-gpu==1.14** (probada al 10/10/2019, en GPUs NVIDIA)
- **tensorflow==1.14** (probada al 10/10/2019, en CPUs Intel)

En caso de querer utilizar TensorFlow con una GPU, se necesita seguir esta guía de instrucciones: [https://www.tensorflow.org/install/gpu#pip\\_package](https://www.tensorflow.org/install/gpu#pip_package)

Básicamente es una guía que prepara nuestro sistema para utilizar GPUs de Nvidia (en las cuales se pueden aprovechar los CUDA cores, una tecnología propia de Nvidia, para mejorar el rendimiento).

En caso de querer utilizar solamente el CPU, no hace falta más que instalar TensorFlow y aquellos paquetes de Python que son necesarios para que TensorFlow haga su magia.

### **NumPy**

NumPy es un paquete que le agrega a Python mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.

Es una de las librerías más utilizadas. Obviamente, TensorFlow hace gran uso de NumPy.

### **Pillow**

Pillow es un fork de PIL (Python Imaging Library), una biblioteca gratuita para Python que agrega soporte para abrir, manipular y guardar muchos formatos de archivo de imagen diferentes.

Al estar trabajando con imágenes, nos resulta útil.

### **SciPy**

SciPy es una biblioteca que se compone de herramientas y algoritmos matemáticos.

### **Keras**

Keras es una biblioteca de Redes Neuronales de Código Abierto escrita en Python. Es capaz de ejecutarse sobre TensorFlow. Está especialmente diseñada para posibilitar la experimentación en más o menos poco tiempo con redes de Aprendizaje Profundo

En nuestro proyecto, utilizamos TensorFlow, quien es el que realiza las llamadas a Keras.

## **Repositorios**

- Repositorio de AMP Tech que contiene el código original del que nos basamos: <https://github.com/puigalex/AMP-Tech/blob/master/CNN%20desde%20cero/>
- Repositorio actual de nuestro proyecto, con el código comentado para entender su funcionamiento: <https://github.com/Squiro/red-neuronal>