ANEXO 1. ENTORNO DE EJECUCIÓN

El lenguaje de programación que se utiliza en el presente libro es Python, el cual puede ser ejecutado en diferentes entornos de ejecución como distribuciones basadas en Conda[27] y notebooks tipo Jupyter[28]. Estos entornos de ejecución permiten la inclusión de frameworks como Tensorflow[29], OpenCV[30] o Keras[31], que permiten la definición de modelos de aprendizaje automático y el procesamiento de imágenes.

Miniconda es una distribución libre y liviana para conda. Es una versión sencilla de Anaconda que incluye sólo conda, Python, los paquetes de los que dependen, y un pequeño número de otros paquetes útiles, incluyendo pip, zlib, entre otros. Sin embargo, mediante el comando "*conda install"* es posible instalar más de 720 paquetes adicionales desde el repositorio de Anaconda[32].

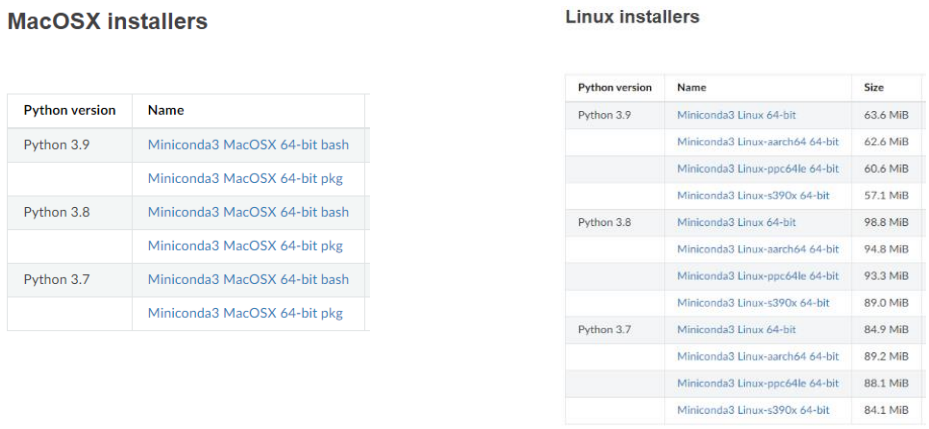
Latest Miniconda Installer Links

Platform	Name	SHA
Windows	Miniconda3 Windows 64-bit	b33
	Miniconda3 Windows 32-bit	24f
MacOSX	Miniconda3 MaxOSX 64-bit bash	786
	Miniconda3 MaxOSX 64-bit pkg	8fa
Linux	Miniconda3 Linux 64-bit	1ea
	Miniconda3 Linux-aarch64 64-bit	487
	Miniconda3 Linux-ppc64le 64-bit	fa9
	Miniconda3 Linux-s390x 64-bit	1fa

Windows installers

Python version	Name	Size
Python 3.9	Miniconda3 Windows 64-bit	58.1 MiB
Python 3.8	Miniconda3 Windows 64-bit	57.3 MiB
Python 3.7	Miniconda3 Windows 64-bit	55.8 MiB
Python 3.9	Miniconda3 Windows 32-bit	55.3 MiB
Python 3.8	Miniconda3 Windows 32-bit	54.5 MiB
Python 3.7	Miniconda3 Windows 32-bit	55.3 MiB

[27] https://github.com/conda/conda
[28] https://jupyter.org
[29] https://www.tensorflow.org
[30] https://opencv.org
[31] https://keras.io
[32] https://conda.io/en/latest/miniconda.html



MacOSX installers

Python version	Name
Python 3.9	Miniconda3 MacOSX 64-bit bash
	Miniconda3 MacOSX 64-bit pkg
Python 3.8	Miniconda3 MacOSX 64-bit bash
	Miniconda3 MacOSX 64-bit pkg
Python 3.7	Miniconda3 MacOSX 64-bit bash
	Miniconda3 MacOSX 64-bit pkg

Linux installers

Python version	Name	Size
Python 3.9	Miniconda3 Linux 64-bit	63.6 MiB
	Miniconda3 Linux-aarch64 64-bit	62.6 MiB
	Miniconda3 Linux-ppc64le 64-bit	60.6 MiB
	Miniconda3 Linux-s390x 64-bit	57.1 MiB
Python 3.8	Miniconda3 Linux 64-bit	98.8 MiB
	Miniconda3 Linux-aarch64 64-bit	94.8 MiB
	Miniconda3 Linux-ppc64le 64-bit	93.3 MiB
	Miniconda3 Linux-s390x 64-bit	89.0 MiB
Python 3.7	Miniconda3 Linux 64-bit	84.9 MiB
	Miniconda3 Linux-aarch64 64-bit	89.2 MiB
	Miniconda3 Linux-ppc64le 64-bit	88.1 MiB
	Miniconda3 Linux-s390x 64-bit	84.1 MiB

Figura 42. Ejemplos de opciones de instalación de *miniconda* para diferentes sistemas operativos.

En su repositorio[33], es posible descargar instaladores para Windows (32 y 64 bits), MacOSX (64 bits) y Linux (64 bits), con la posibilidad de seleccionar entre las últimas versiones de Python. Una vez se instala y ejecuta la aplicación, es posible ejecutar también *jupyter notebook* (Figuras A1.2 y A1.3).

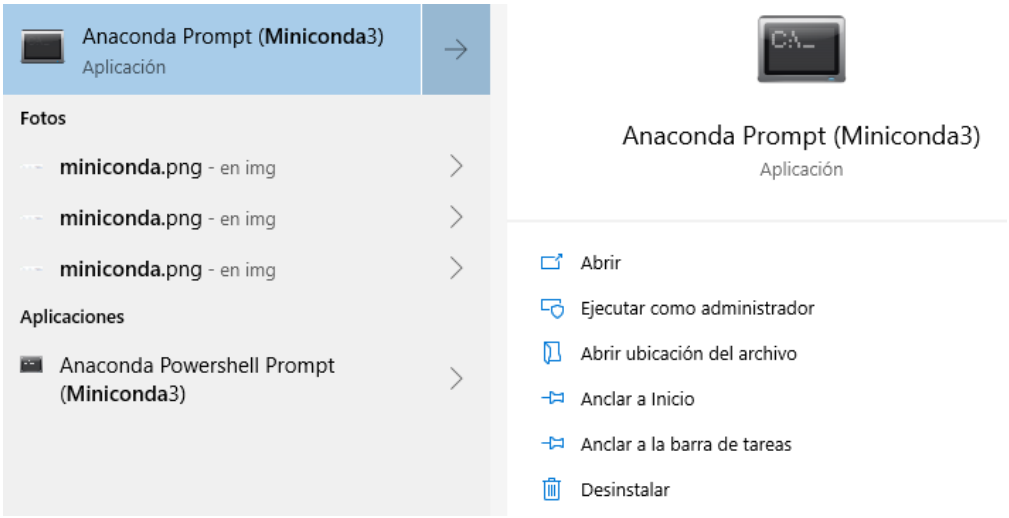


Figura 43. Ejecución de *miniconda* en Windows.

[33] https://conda.io/en/latest/miniconda.html

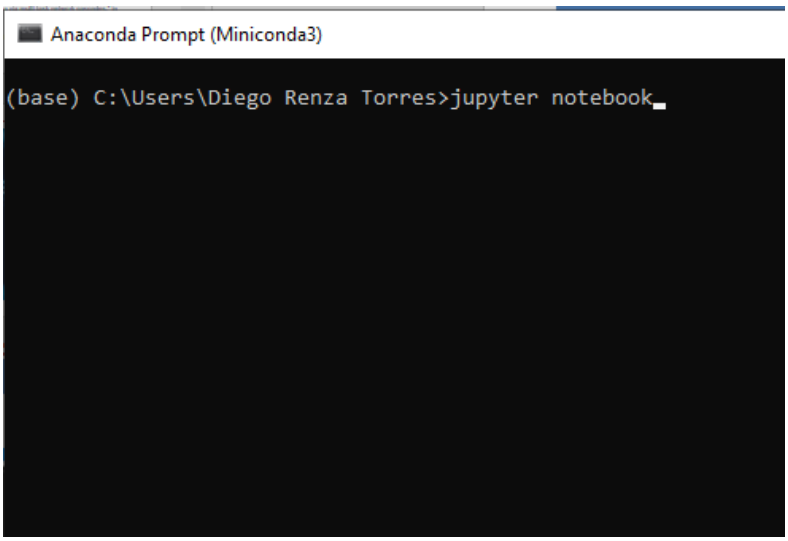


Figura 44. Ejecución de *jupyter notebook* en Windows.

Al ejecutar *jupyter notebook* se lanza el navegador web predeterminado del sistema accediendo al computador local a través de la interfaz de red loopback (localhost) en el puerto predeterminado 8888 y mostrando el árbol de carpetas y archivos del sistema (Figura A1.4). En el contexto de ejecución del *jupyter notebook*, los archivos principales tendrán extensión ipynb, como se muestra en la Figura A1.5.

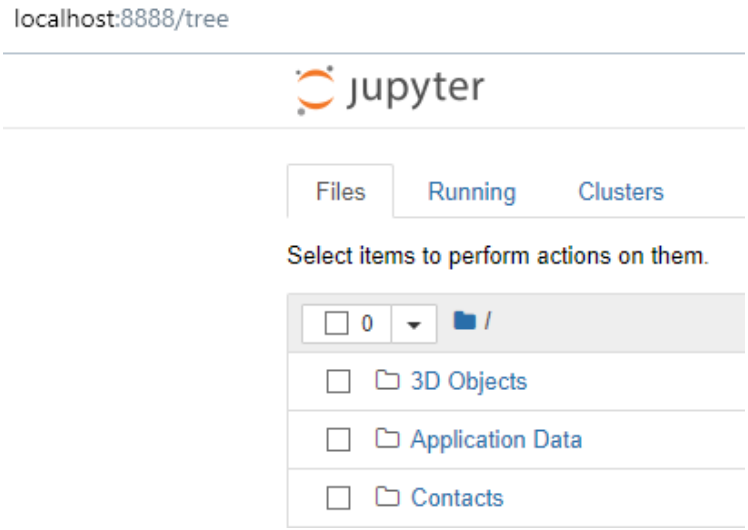


Figura 45. Ambiente de trabajo en *jupyter notebook.*

Figura 46. Ejemplos de archivos creados con *jupyter notebook.*

Los archivos ipynb se caracterizan por incluir dos tipos de bloques principales: bloques de texto y bloques de código (Figura A1.6). En los bloques de texto es posible incluir explicaciones, ecuaciones, figuras, divisiones de sección, entre otras que faciliten el contexto del código contenido en el bloc de notas. En cuanto a los bloques de código, su característica principal es la de posibilitar la ejecución tanto de líneas seleccionadas, como de bloques independientes (bloque a bloque) o ejecutar todo el contenido del bloc de notas (Figura A1.7).

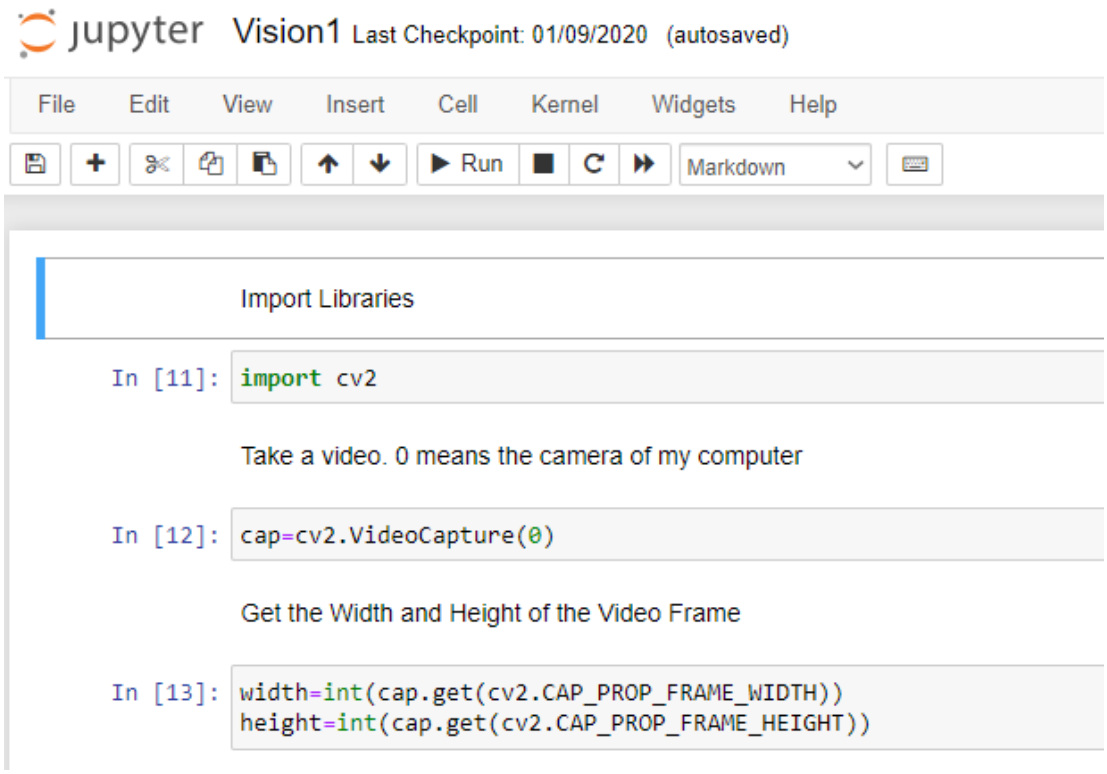


Figura 47. Ejemplos de bloques de código en un *jupyter notebook.*

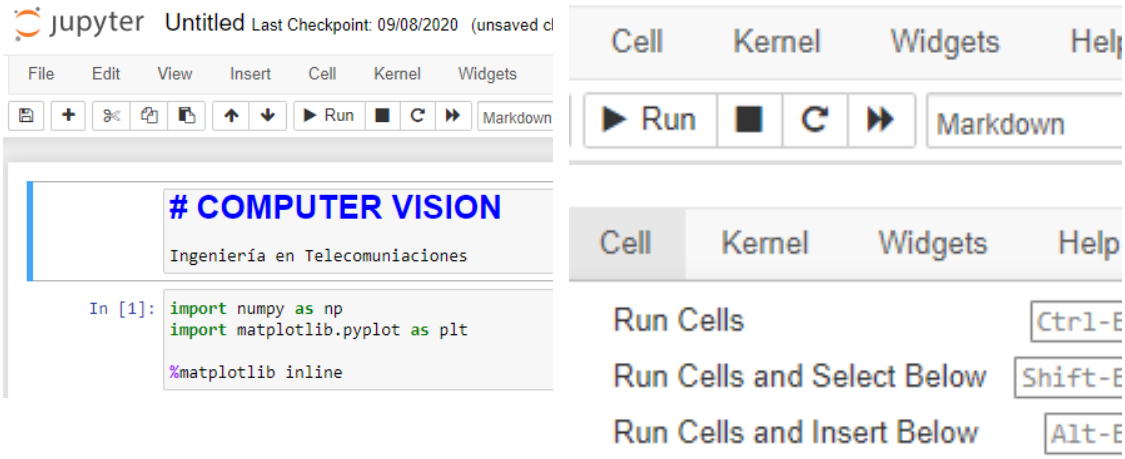


Figura 48. Opciones de ejecución de código en un *jupyter notebook.*

A manera de ejemplo, la Figura A.1.8 muestra varios bloques de código en el cual se importa Tensorflow, se crea un vector de 12 elementos con los números del 0 al 11, y luego se aplica un operador de multiplicación. Cada bloque se ejecutó de manera independiente, donde los números entre corchetes indicados en la parte izquierda indican el orden de ejecución de estos. Adicionalmente, y de acuerdo al tipo de operación realizada, es posible visualizar el resultado de las líneas ejecutadas, como lo muestra las salidas de los bloques 3 y 4.

```
In [2]: import tensorflow as tf

In [3]: x = tf.range(12)
        x

Out[3]: <tf.Tensor: shape=(12,), dtype=int32, numpy=array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])>

In [4]: x*2

Out[4]: <tf.Tensor: shape=(12,), dtype=int32, numpy=array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18, 20, 22])>

In [ ]:
```

Figura 49. Bloques de código en un *jupyter notebook.*

Existen herramientas para ejecución de *jupyter notebook* en servicios alojados en la nube. Una de las más utilizadas en la actualidad es Google CoLaboratory, el cual es un proyecto de investigación de Google creado para ayudar a difundir la educación y la investigación del aprendizaje automático. Es un entorno de *Jupyter notebook* que no requiere ninguna configuración para su uso y se ejecuta

completamente en la nube[34]. Por lo tanto, Colaboratory (conocido simplemente como Colab), permite escribir y ejecutar código Python en un navegador web, incluyendo además acceso gratuito a una GPU y facilitando el trabajo compartido de estos archivos[35].

Para el uso de Colab, es necesario agregarlo a las aplicaciones del entorno G-suite (https://colab.research.google.com). Posterior a su instalación y ejecución, se agrega una carpeta en Google drive diseñada para repositorio de estos blocs de notas, aunque es posible almacenarlos y ejecutarlos desde cualquier carpeta de este drive.

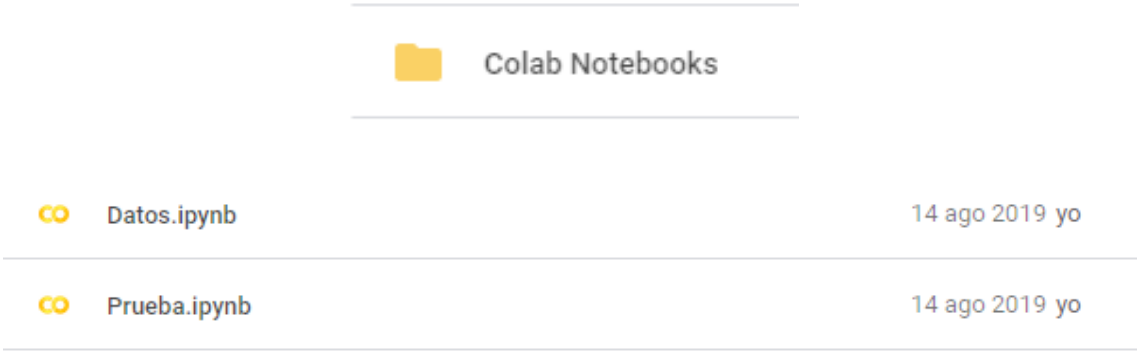


Figura 50. Estructura de archivos de CoLaboratory en Google drive.

Figura 51. Entorno de ejecución de un notebook en *CoLaboratory.*

[34] https://research.google/tools/
[35] https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=5fCEDCU_qrC0

Para ampliar la comprensión de las funcionalidades de *jupyter notebook*, se sugiere al lector la consulta del siguiente enlace:

https://nbviewer.jupyter.org/github/jupyter/notebook/tree/master/docs/source/examples/Notebook/

