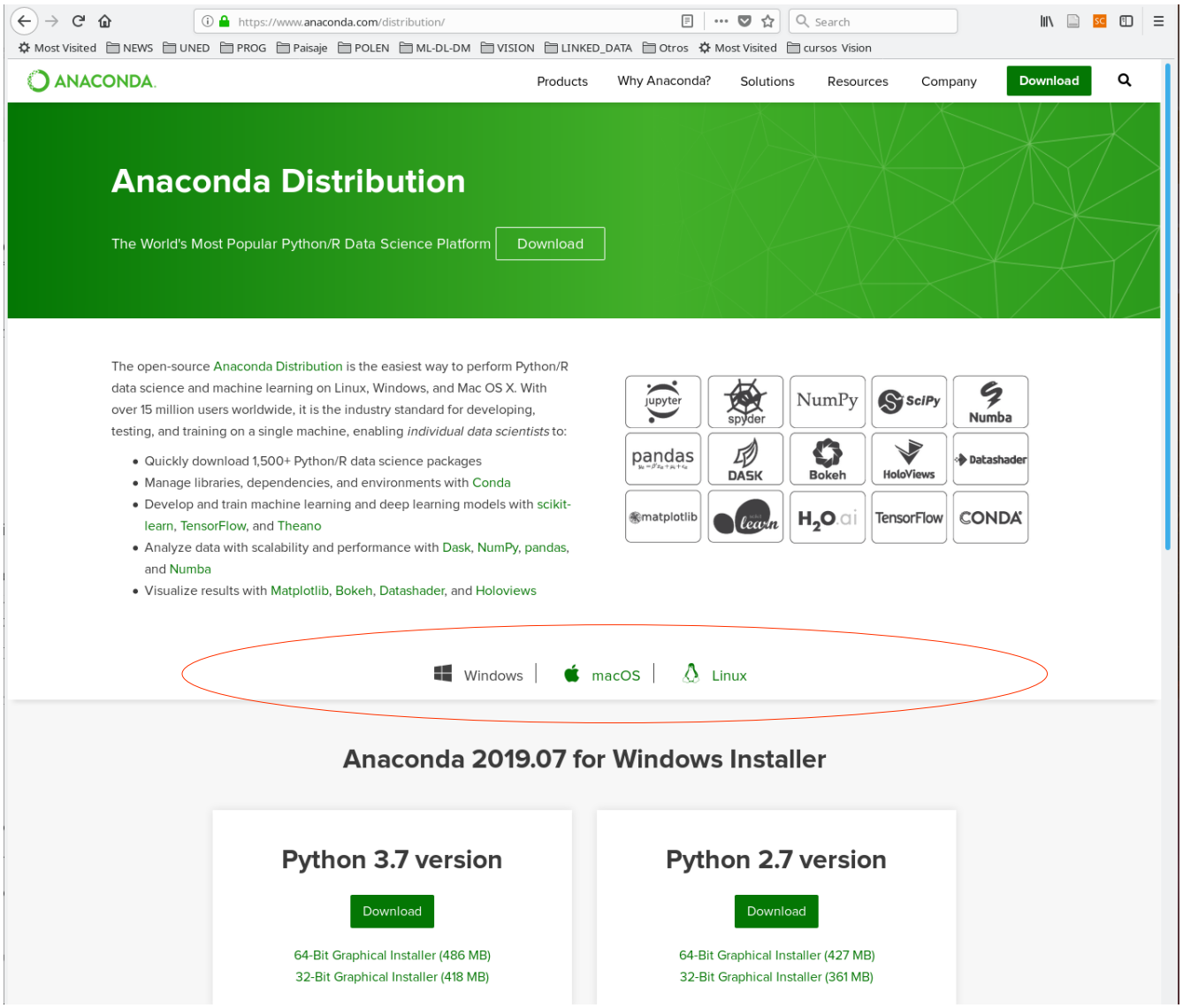
ANACONDA: Pasos para crear entorno virtual

1) Descargar el instalador adecuado para el sistema operativo:

https://www.anaconda.com/distribution/



Tened en cuenta que para que funcione anaconda, es conveniente que esté en el path:

“By default, this installer modifies your bash profile to activate the base environment of Anaconda3 when your shell starts up. To disable this, choose "Customize" at the "Installation Type" phase, and disable the "Modify PATH" option. If you decline this option, the executables installed by this installer will not be available on PATH. You will need to use the full executable path to run commands, or otherwise initialize the base environment of Anaconda3 on your own. “

2) Una vez instalado anaconda deberéis construir el entorno virtual para visión. La gestión se puede realizar desde terminal o utilizando la aplicación anaconda navigator (más visual pero con menor control). Se recomienda hacer lo siguiente:

Gestión desde terminal
======================

1. Paso 1: # actualizar conda: **conda update -n base -c defaults conda**

2. Paso 2: # importar el entorno virtual creado para el curso:
   conda env create -f CV.yml -n ComputerVision

   # para crear fichero para exportar entorno:
   conda env export > CV.yml

   # Para crear entorno por defecto:
   conda create --name CV python=3.7

3. Paso 3: Activar el nuevo entorno creado:

   # activar entorno: **conda activate ComputerVision**
   # desactivar entorno: **conda deactivate**
   # entornos disponibles y cual está activo: **conda info --envs**

4. Instalar paquetes adicionales:

   # si se quieren instalar otros paquetes (se puede controlar la versión que se instala). Ejemplo:
   conda install opencv

   # **los paquetes también se pueden instalar con pip**, pero ojo, al ser gestores distintos, los paquetes instalados con pip no aparecerán en anaconda navigator.

   PIP environment
   ===============
   # instalar último paquete: **pip install packageName**
   # instalar paquete específico: **pip install packageName==versionNumber**
   # Preguntar versiones existentes: **pip install packageName==**

   # importar environment
        (<env_name>)$ **pip install -r path/to/requirementsPIP.txt**

   # para exportar environment: **pip freeze > requirementsPIP.txt**

Gestión desde anaconda-navegator
================================
* En submenú "Environments"
   - se puede gestionar la configuración de entornos virtuales
   - se pueden instalar paquetes de la distribución actual de anaconda

* Desde submenú "Home"- se instalan y ejecutan tanto Spyder como Jupyter-notebook. Si la aplicación no está instalada en el entorno, el botón muestra "install", si ya está instalado "Launch").

Spyder3

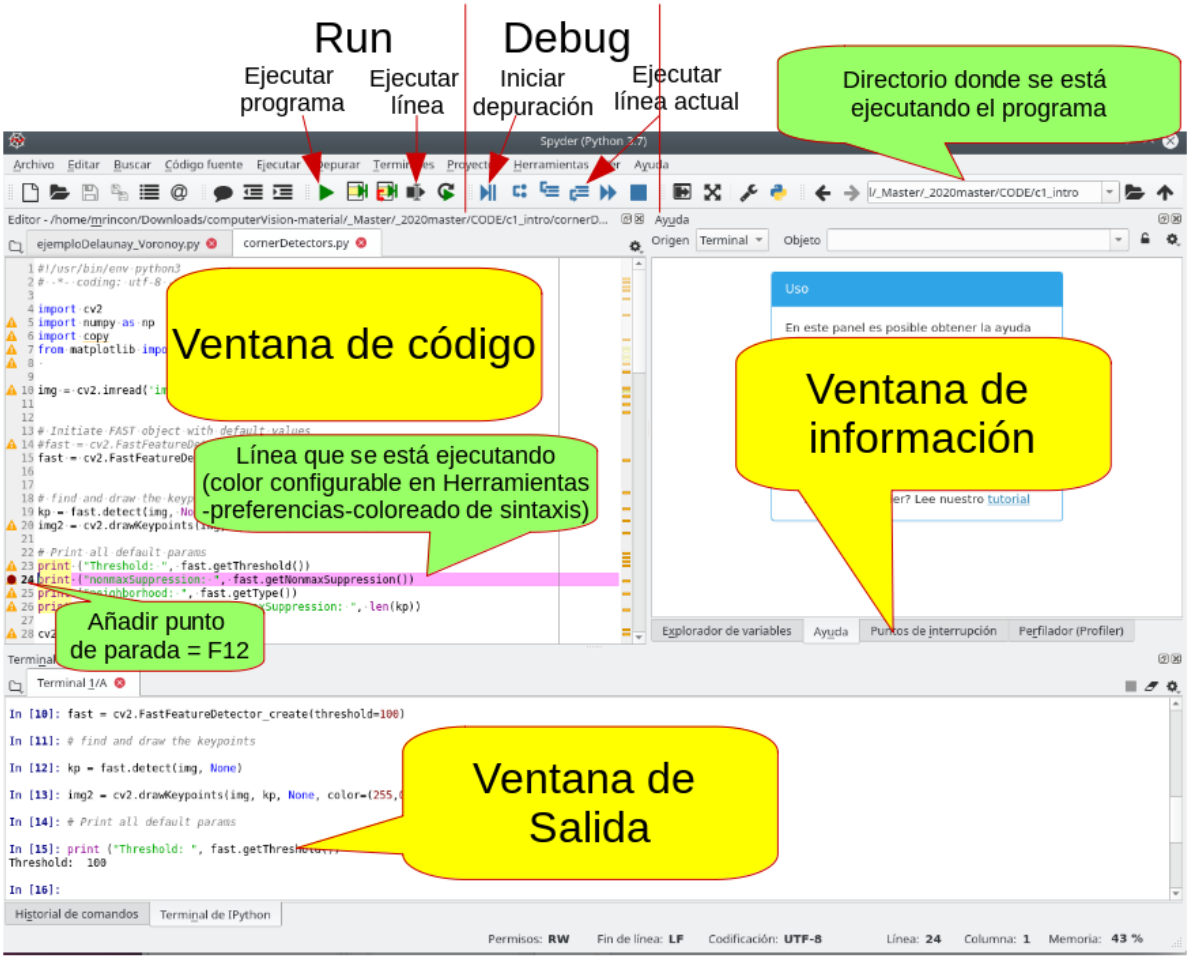


Fig. 2: GUI del entorno de programación con Spyder3.

La figura 2 muestra el entorno de programación Spyder3. Se supone que ya se sabe lo que es un entorno de programación/depuración, por lo que en la figura se resumen los puntos más importantes.

1) Es de especial relevancia configurar bien el directorio de trabajo donde se está ejecutando el programa, sobre todo si se accede al directorio para leer o escribir datos a ficheros.

2) La depuración se puede realizar de dos maneras:

   1) Con los botones que hemos denominado modo “Run”, que permiten ejecutar líneas sueltas de código (ejecutar línea) o ejecutar el programa completo (ejecutar programa). No se tienen en cuenta los puntos de parada (breakpoints).

   2) Con los botones que hemos denominado modo “Debug”, que permiten la ejecución paso a paso teniendo en cuenta los puntos de parada. La línea actual queda resaltada en color y se puede ejecutar el código paso a paso, entrar en una función, salir de ella, continuar, etc.

3) En la ventana de salida también se pueden ejecutar comandos de Python durante la depuración de un programa.

3) A veces, el núcleo del depurador se queda bloqueado y es necesario reiniciarlo:
Menú “Terminales” + Reiniciar núcleo

4) Para configurar PYTHONPATH (para acceder a los directorios donde están organizados los módulos de python propios), acceder en
Menú “Herramientas” + Administrador del PYTHONPATH.

5) La personalización del entorno se realiza a través de:
Menú “Herramientas” + Preferencias.

6) No se puede mezclar caracteres de indentación. Se puede elegir entre Tabulaciones y Espacios. Se configura en:
Menú “Preferencias” + Editor + Opciones avanzadas + Caracteres de indentación

Jupyter-notebook

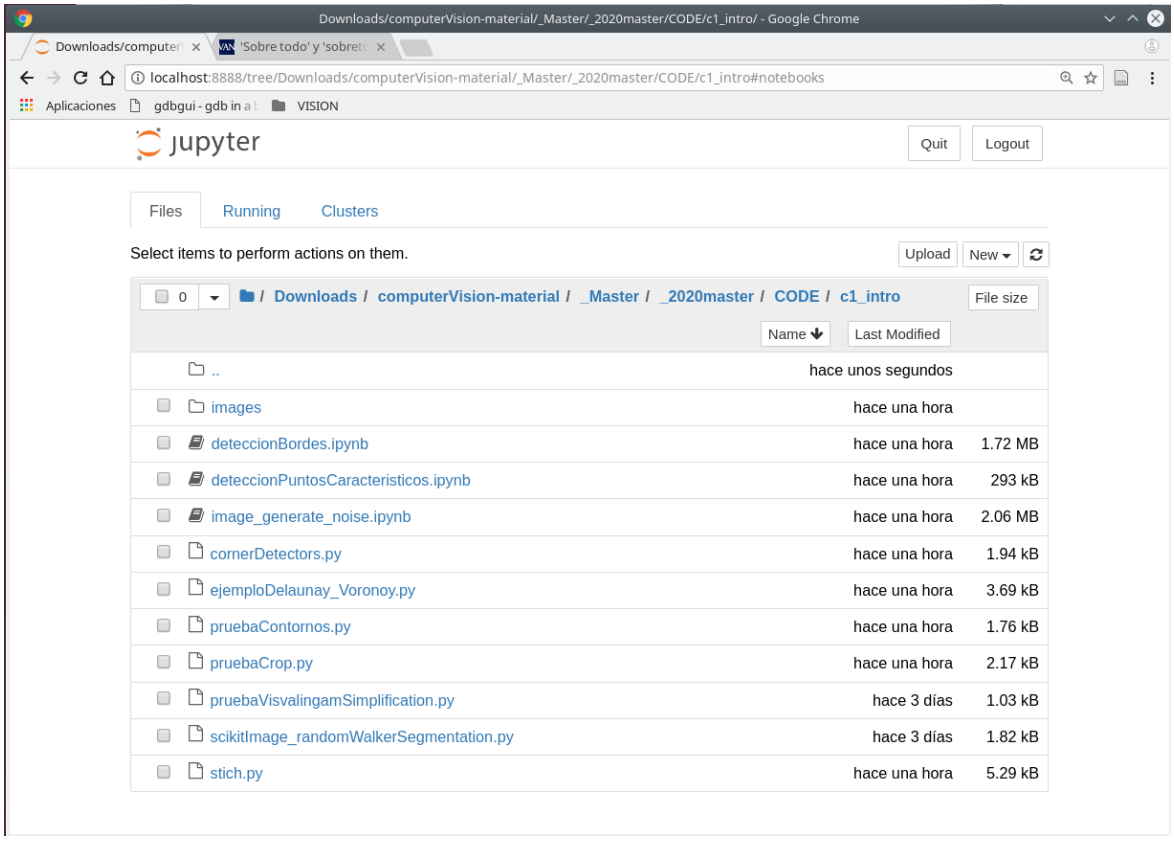


Fig. 3: Ventana inicial GUI Jupyter-notebook.

La figura 3 muestra la ventana inicial del entorno Juputer-notebook. Desde esta ventana se navega hasta acceder a los ficheros de código (.ipynb). Una vez

seleccionado el fichero, se entra en el modo edición (figura 4). También se puede editar un nuevo programa (Botón “New” + seleccionar “Python3”).

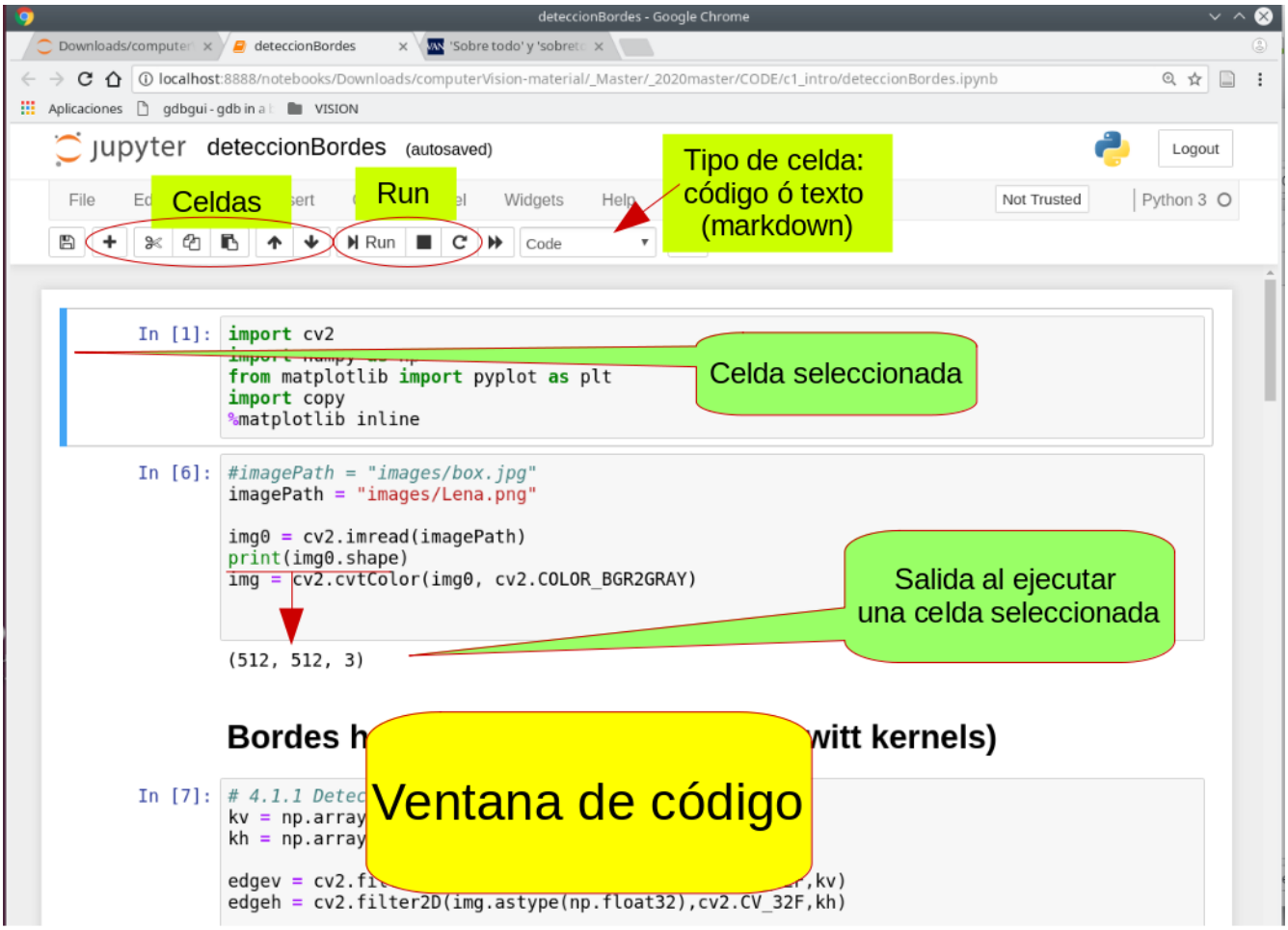


Fig. 4: Ventana edición y depuración de Jupyter-notebook.

La edición se realiza por bloques (celdas) y éstas pueden ser tanto de código como de texto. Las celdas de código se pueden ejecutar y la salida, tanto si es texto como si son imágenes o gráficos, se presentan a continuación de la celda. Las celdas de texto se pueden formatear (tamaño de letra, …). Esto permite documentar muy bien un programa, pues permite combinar la estructura del documento, explicaciones puntuales, código y salida del programa.

Las celdas parten el código en apartados independientes pero es conveniente estructurar el código en el mismo orden que se haría en un programa python normal (.py). Se recomienda inspeccionar los ejemplos enviados.