

PRÁCTICAS PROGRAMACIÓN TÉCNICA Y CIENTÍFICA

GRADO INGENIERÍA INFORMÁTICA. UNIVERSIDAD DE GRANADA

SESIÓN 1. INSTALACIÓN E INICIO

1. Configuración del entorno

Con la finalidad de tener un entorno homogéneo para todos los estudiantes, las prácticas se van a realizar en Ubuntu 20.04 con la distribución científica Anaconda que incluye Python 3.9. Se pone a disposición de los estudiantes un disco de máquina virtual **ptevi.dvi** con Ubuntu 20.04 y con la distribución Anaconda ya instalada.

1.1. Instalación en la máquina virtual

Pasos para la creación de la máquina virtual.

Descarga en <https://www.virtualbox.org/> de Virtual Box 6.1. Instalamos.

Descarga de VM VirtualBox Extension Pack. Instalamos igual versión de Virtual Box.

Obtener el disco de máquina virtual **ptevi.dvi** de Prado y crear una máquina Linux Ubuntu 64 bits con al menos 4 GB de RAM. El usuario root es “aulas” con contraseña “aulas”.

Obtener la siguiente versión de la distribución de Anaconda de Anaconda.com [Anaconda3-2021.11-Linux-x86_64.sh](#) o bien la localizamos en la carpeta de descargas de la máquina virtual. Instalar.

Si se descarga el fichero de Anaconda.com tendremos que cambiar los permisos para que se pueda ejecutar. Abrimos la terminal en la carpeta de “descargas” y establecemos todos los permisos con `chmod u+x nombreFichero`. Poniendo luego `./nombreFichero` se puede ejecutar e instalar siguiendo las instrucciones. Dejar carpetas por defecto.

Importante: cuando te diga si quieres **inicializar Anaconda3** pon “yes” en la instalación. **Cerramos esa terminal** y abrimos otra. Probamos el intérprete de python y vemos la versión con `python --version` o `python3 --version`

Obtener el simulador **CoppeliaSim V4.4.0 rev0 for Ubuntu 20.04 versión EDU** que usaremos más adelante de Prado. Instalar simplemente descomprimiendo el fichero en el home.

Características opcionales de la máquina virtual. La máquina suministrada contiene los drivers adecuados para funcionar en los laboratorios de la Escuela pero si preferís mejorar dichos drivers y adaptarlos a vuestros portátiles hay que ejecutar las Guest Additions.

1.2. Pasos iniciales con Anaconda3

En una instalación de Ubuntu 20.04 tenemos instalado por defecto python 3.8.2 pero en nuestra distribución de Anaconda tendremos Python 3.9.7. y Spyder IDE 5.1.5.

El entorno Anaconda se activa siempre por defecto al abrir un terminal debido a que se puso “yes” en la instalación cuando se preguntó por la inicialización. Este entorno se puede desactivar con “conda deactivate” y volver a activar con “conda activate”.

Si ponemos “conda config --set auto_activate_base false” ya no se activa de forma automática al abrir un terminal y habrá que activarlo manualmente. Para volver a la situación anterior poned: “conda config --set auto_activate_base true”

Ahora accedemos a <https://docs.anaconda.com/anaconda/user-guide/getting-started/>

donde explica que podemos usar tanto una interfaz de órdenes (conda) como una gráfica (anaconda-navigator). Se pueden instalar paquetes adicionales usando las dos interfaces pues se gestiona correctamente.

Probamos primero Anaconda Navigator y desde él usaremos spyder y jupyter notebook. Lanzamos spyder y creamos nuestro programa de prueba “hola mundo”. Cerramos spyder. Hacemos lo mismo pero usando Jupyter Notebook. Existen muchas plantillas de ejemplos para Jupyter Notebook. Buscar jupyter notebook gallery.

Ahora cerramos Anaconda Navigator y trabajaremos desde una terminal en linux. Desde esa ventana se puede ejecutar:

python → la manera más rudimentaria de trabajar

ipython → tiene una interfaz mejorada

jupyter qtconsole → tiene posibilidades gráficas

Desde el terminal se puede lanzar también: spyder o bien jupyter-notebook

2. Primeros pasos con python

Trabajaremos primero con las herramientas más rudimentarias hasta llegar al entorno que usaremos finalmente.

2.1. El intérprete de Python

La manera más rudimentaria sería usar directamente el intérprete de python sin más. Para ellos lanzamos una terminal en linux o Anaconda prompt en windows y escribimos “python”.

Seguir la instrucciones del profesor para realizar: Operaciones sencillas Asignar a=5, b=7 c=a+b

Uso de la orden print. Ejemplo print(a), print(“Hola mundo”).

Salimos con exit() o ^d

Ahora probamos los mismos ejemplos en ipython para comprobar que hay cierta mejora.

2.2. Usar un editor de texto para crear un programa sencillo.

Los ficheros con código fuente en python usan la extensión .py En linux podemos usar gedit y salvarlo con nombre prueba.py y ejecutarlo con python prueba.py

Script ejecutable en entornos Linux: Podemos convertirlo en ejecutable pero para ello tenemos que añadir en la primera línea qué intérprete de python tiene que usar con: `#!/usr/bin/env python`

El fichero podría ser algo así:

```
#!/usr/bin/env python
```

```
print("hola mundo")
```

luego establecemos los permisos de ejecución `chmod a+x prueba.py`

2.3. El editor spyder

Ahora lanzamos spyder desde una terminal. Probamos algunos de los programas anteriores.

Operaciones sencillas Asignar `a=5, b=7 c=a+b`

Uso de la orden print. Ejemplo `print(a)`, `print("Hola mundo")`.

Ahora podemos depurar el programa con el depurador y la ejecución paso a paso. Podemos reiniciar el núcleo desde la ventana del terminal (reset desde el terminal ipython). Podemos ver todas las variables definidas con `locals()` desde el terminal.

3. Implementar estos ejercicios en python sin utilizar funciones

- Realizar un programa que solicite la base y altura de un triángulo y calcule su área mostrando la salida por pantalla.
- Realizar un programa que solicite el radio de una circunferencia y calcule su longitud y el área del círculo mostrando la salida por pantalla.
- Realizar un programa que solicite los dos catetos de un triángulo rectángulo y que calcule la hipotenusa mostrando la salida por pantalla. (Usad `math.sqrt()` para la raíz cuadrada).