

U.T. 6: Programación de scripts

Instrucciones de instalación APP

Administración de SS00

Instalación

NOTA: Toda la instalación se deberá realizar con **root**.

En primer lugar, nos descargaremos la herramienta **git** para bajarnos el proyecto ubicado en *GitHub*:

```
root@C02-16:~# apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Ilustración 1 Instalar la herramienta git

A continuación, descargaremos el proyecto desde el repositorio de *GitHub* de la siguiente manera: [Repositorio en GitHub](#):

```
root@C02-16:~# git clone https://github.com/RubGomCoding/LinuxProject
Cloning into 'LinuxProject'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), 4.72 MiB | 1.06 MiB/s, done.
```

Ilustración 2 Descargar proyecto

Hecho esto, entraremos dentro de la carpeta del proyecto y descomprimiremos el fichero nombrado *FINAL.zip*, nosotros utilizaremos la utilidad **unzip**:

```
root@C02-16:~# cd LinuxProject/
root@C02-16:~/LinuxProject# ls
FINAL.zip  README.md
root@C02-16:~/LinuxProject# unzip FINAL.zip
Archive:  FINAL.zip
  creating: FINAL/
  extracting: FINAL / nottione
```

Ilustración 3 Descomprimir archivo proyecto

Una vez lo hayamos descomprimido, deberemos ver algo similar a lo siguiente dentro del directorio

```
root@C02-16:~/LinuxProject/FINAL# ls
AppConfigurations  Gui.py      Utils.py      entorno-pruebas  nombre_usuarios.txt  tree_view.py
Assets              TaskCront.py  __pycache__  fichero_cron.txt  requirements.txt
```

Ilustración 4 Vista directorio proyecto

Administración de SSOO

Instrucciones de instalación APP

Hecho esto, deberemos crear un **entorno virtual**. Esto lo haremos con estos 3 simples pasos:

- Instalar herramienta para crear entornos virtuales → **sudo apt install python3-venv**
- Crear entorno virtual → **python3 -m venv .venv**
- Entrar dentro del entorno virtual → **source .venv/bin/activate**

En WSL los pasos se verían del siguiente modo:

```
root@C02-16:~/LinuxProject/FINAL# sudo apt install python3-venv
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-venv is already the newest version (3.12.3-0ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@C02-16:~/LinuxProject/FINAL# python3 -m venv .venv
root@C02-16:~/LinuxProject/FINAL# source .venv/bin/activate
```

Ilustración 5 Pasos WSL

Si lo hemos hecho bien, veremos que tenemos un (.venv) en la **prompt**, lo que indica que estamos dentro de un *entorno virtual*.

```
(.venv) root@C02-16:~/LinuxProject/FINAL#
```

Ilustración 6 Prompt entorno virtual

Ahora, ya tenemos un entorno totalmente **aislado** para instalar nuestras librerías y lanzar nuestro programa. Las librerías, las instalaremos gracias al fichero **requirements.txt**, en el cual se encuentran todas las librerías necesarias para lanzar el programa.

```
(.venv) root@C02-16:~/LinuxProject/FINAL# cat requirements.txt
babel==2.16.0
backup.py==0.1.1
certifi==2025.1.31
charset-normalizer==3.4.1
colorama==0.4.6
customtkinter==5.2.2
darkdetect==0.8.0
docopt==0.6.2
idna==3.10
packaging==24.2
pillow==11.1.0
pipreqs==0.4.13
requests==2.32.3
tkcalendar==1.6.1
tqdm==4.67.1
urllib3==2.3.0
yarg==0.1.10
```

Ilustración 7 Contenido requirements.txt

Administración de SSOO

Instrucciones de instalación APP

Este fichero lo hemos creado gracias a la utilidad **freeze** del siguiente modo (no tenéis que crearlo vosotros):

```
(.venv) root@C02-16:~/LinuxProject/FINAL# pip3 freeze > requirements.txt
```

Ilustración 8 Creación requirements.txt

El fichero requirements.txt lo instalaremos del siguiente modo **pip install -r requirements.txt**:

```
(.venv) root@C02-16:~/LinuxProject/FINAL# pip install -r requirements.txt
Requirement already satisfied: asttokens==3.0.0 in ./venv/lib/python3.12/site-packages (from -r requirements.txt (line 1)) (3.0.0)
Requirement already satisfied: attrs==25.1.0 in ./venv/lib/python3.12/site-packages (from -r requirements.txt (line 2)) (25.1.0)
Requirement already satisfied: backcall==0.2.0 in ./venv/lib/python3.12/site-packages (from -r requirements.txt (line 3)) (0.2.0)
Requirement already satisfied: beautifulsoup4==4.13.3 in ./venv/lib/python3.12/site-packages (from -r requirements.txt (line 4)) (4.13.3)
```

Ilustración 9 Instalación librerías

Y, estas no son todas las librerías, la librería de **tkinter** que es con la que se levanta toda la interfaz gráfica se instalará con **apt** del siguiente modo **sudo apt-get install python3-tk**:

```
(.venv) root@C02-16:~/LinuxProject/FINAL# sudo apt-get install python3-tk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  blt libtcl8.6 libtk8.6 libxft2 libxss1 tk8.6-blt2.5
Suggested packages:
  blt-demo tcl8.6 tk8.6 tix python3-tk-dbg
The following NEW packages will be installed:
  blt libtcl8.6 libtk8.6 libxft2 libxss1 python3-tk tk8.6-blt2.5
```

Ilustración 10 Instalar librería Tkinter

Lanzamiento del proyecto

Ahora, para lanzar el proyecto, simplemente desde la raíz del proyecto, lanzaremos la orden: **.venv/bin/python3 Gui.py** la cual simplemente ejecuta el programa con *Python 3.12.3*.



Ilustración 11 Ejecución programa