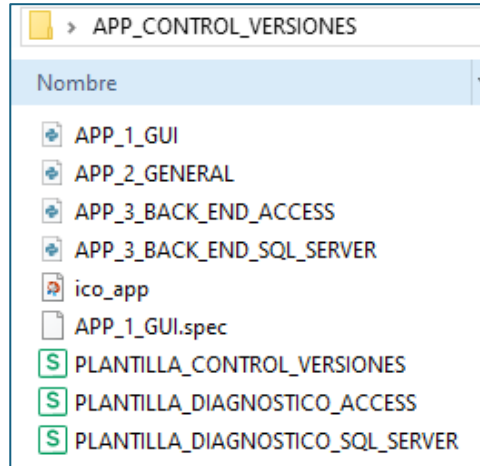


MANUAL PARA COMPILAR EL APP EN EJECUTABLE EN WINDOWS

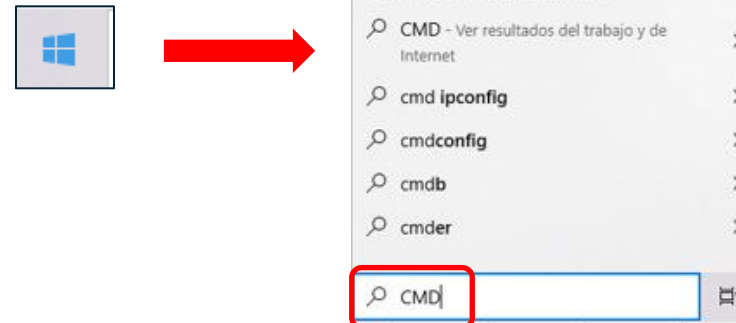
Como paso previo, el pc debe tener **Python 3.7** instalado. En este manual no se explica como realizar esta instalación (en youtube existen múltiples videos gratuitos que ayudan a ello).

Paso 1. Crear una ubicación donde guardar los archivos siguientes (esta ruta en adelante se llamará mi_ruta).



Paso 2. Crear el entorno virtual donde instalar los frameworks Python necesarios.

Abrimos la consola MS-DOS:



- Entramos en la ubicación de mi_ruta

Escribimos **cd mi_ruta**

- Creamos el entorno virtual (se puede llamar como se quiera aquí lo llamo mi_entorno_virtual)

Escribimos **python -m venv mi_entorno_virtual**

- Activamos el entorno virtual

Escribimos **mi_entorno_virtual\Scripts\activate.bat**

```

C:\> Símbolo del sistema

Microsoft Windows [Versión 10.0.19045.5247]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\user> cd C:\Users\user\Desktop\APP_CONTROL_VERSIONES
C:\Users\user\Desktop\APP_CONTROL_VERSIONES> python -m venv mi_entorno_virtual
C:\Users\user\Desktop\APP_CONTROL_VERSIONES> mi_entorno_virtual\Scripts\activate.bat
(mi_entorno_virtual) C:\Users\user\Desktop\APP_CONTROL_VERSIONES>
```



Esto indica que el entorno virtual esta activado.

Paso 3. Instalamos los frameworks Python necesarios (los que no vienen instalados con Python) para compilar el app.

Escribimos:

pip install numpy==1.22.0 pandas==1.5.0 pyinstaller==4.7 pyodbc==4.0.32 thread6==0.2.0 xlwings==0.27.15

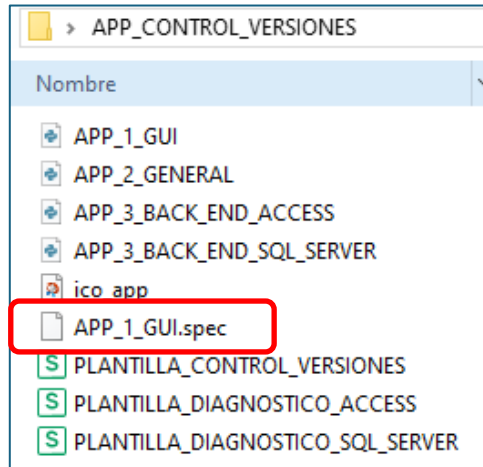
En caso de no ser administrador del equipo añadir **--user** al final:

pip install numpy==1.22.0 pandas==1.5.0 pyinstaller==4.7 pyodbc==4.0.32 thread6==0.2.0 xlwings==0.27.15 --user

```
(mi_entorno_virtual) C:\Users\user\Desktop\APP_CONTROL_VERSIONES>pip install numpy==1.22.0 pandas==1.5.0 pyinstaller==4.7 pyodbc==4.0.32 thread6==0.2.0 xlwings==0.27.15
Collecting numpy==1.22.0
  Using cached numpy-1.22.0-cp39-cp39-win_amd64.whl (14.7 MB)
Collecting pandas==1.5.0
  Using cached pandas-1.5.0-cp39-cp39-win_amd64.whl (10.9 MB)
Collecting pyinstaller==4.7
  Using cached pyinstaller-4.7-py3-none-win_amd64.whl (2.0 MB)
Collecting pyodbc==4.0.32
  Using cached pyodbc-4.0.32-cp39-cp39-win_amd64.whl (72 kB)
Collecting thread6==0.2.0
  Using cached thread6-0.2.0-py3-none-any.whl (4.2 kB)
Collecting xlwings==0.27.15
  Using cached xlwings-0.27.15-py3-none-any.whl (1.0 MB)
Collecting pytz>=2020.1
  Downloading pytz-2024.2-py2.py3-none-any.whl (508 kB)
    |████████████████████████████████████████| 508 kB 1.7 MB/s
Collecting python-dateutil>=2.8.1
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
    |████████████████████████████████████████| 229 kB 6.4 MB/s
Collecting altgraph
  Using cached altgraph-0.17.4-py2.py3-none-any.whl (21 kB)
Collecting pefile>=2017.8.1
  Downloading pefile-2024.8.26-py3-none-any.whl (74 kB)
    |████████████████████████████████████████| 74 kB 689 kB/s
```

Esperamos un rato el tiempo que se instalen los frameworks.

Paso 4. Preparamos el fichero .spec que permite compilar el app en ejecutable (se puede abrir con el bloc de notas o Notepad++).



```
# -*- mode: python ; coding: utf-8 -*-

block_cipher = None

a = Analysis(['APP_1_GUI.py'],
             binaries=[],
             datas=[
                 (r"mi_ruta\ico_app.ico", "."),
                 (r"mi_ruta\PLANTILLA_CONTROL_VERSIONES.xlsb", "."),
                 (r"mi_ruta\PLANTILLA_DIAGNOSTICO_ACCESS.xlsb", "."),
                 (r"mi_ruta\PLANTILLA_DIAGNOSTICO_SQL_SERVER.xlsb", ".")
             ],
             hiddenimports=[],
             hookspath=[],
             hooksconfig={},
             runtime_hooks=[],
             excludes=[],
             win_no_prefer_redirects=False,
             win_private_assemblies=False,
             cipher=block_cipher,
             noarchive=False)

pyz = PYZ(a.pure, a.zipped_data,
          cipher=block_cipher)

exe = EXE(pyz,
          a.scripts,
          a.binaries,
          a.zipfiles,
          a.datas,
          [],
          name='APP_CONTROL_VERSIONES_ACCESS_SQL_SERVER',
          debug=False,
          bootloader_ignore_signals=False,
          strip=False,
          upx=True,
          upx_exclude=[],
          runtime_tmpdir=None,
          console=True,
          disable_windowed_traceback=False,
          target_arch=None,
          codesign_identity=None,
          entitlements_file=None)
```

Es el nombre del modulo .py desde el cual se ejecuta el app

Esta lista de tuplas permite incluir los archivos que han de incorporarse al app
→ Reemplazar mi_ruta por la ruta donde se ha creado el entorno virtual

Si se establece a True el app cuando este compilado se abrirá con una consola negra. Si se produce algún error al abrir el app, el error se reflejara en esta consola.

Si pasara lo mas probable es que haya que añadir **hiddenimports**

Una vez realizados los cambios en el fichero .spec, lo guardamos y lo cerramos.

Paso 5. Volvemos a la consola MS-DOS y escribimos:

Pyinstaller APP_1_GUI.spec

```
(mi_entorno_virtual) C:\Users\user\Desktop\APP_CONTROL_VERSIONES>Pyinstaller APP_1_GUI.spec
5253 INFO: PyInstaller: 4.7
5254 INFO: Python: 3.9.5
5378 INFO: Platform: Windows-10-10.0.19045-SP0
5411 INFO: UPX is not available.
5432 INFO: Extending PYTHONPATH with paths
['C:\\Users\\user\\Desktop\\APP_CONTROL_VERSIONES']
11652 INFO: checking Analysis
11654 INFO: Building Analysis because Analysis-00.toc is non existent
11657 INFO: Initializing module dependency graph...
11692 INFO: Caching module graph hooks...
11871 INFO: Analyzing base_library.zip ...
```

Dejamos que se compile el app (puede tardar unos minutos).

La compilación habrá acabado cuando salga esto:

```
530406 INFO: Updating resource type 24 name 1 language 0
530447 INFO: Appending PKG archive to EXE
584271 INFO: Building EXE from EXE-00.toc completed successfully.

(mi_entorno_virtual) C:\Users\user\Desktop\APP_CONTROL_VERSIONES>
```

Paso 6. Recuperamos el app compilado.

En la ruta donde se creo el entorno virtual se habrá creado una subcarpeta llamada **dist** donde se encuentra el app en formato .exe



El app creado aquí se puede usar en cualquier pc sin necesidad de tener Python instalado.

Paso 7. Desactivamos el entorno virtual.

Escribimos **deactivate** en la consola MS-DOS

```
(mi_entorno_virtual) C:\Users\user\Desktop\APP_CONTROL_VERSIONES>deactivate  
C:\Users\user\Desktop\APP_CONTROL_VERSIONES>
```

Paso 8. Eliminar el entorno virtual (opcional)

Escribimos **rmdir mi_entorno_virtual/s** en la consola MS-DOS

```
(mi_entorno_virtual) C:\Users\user\Desktop\APP_CONTROL_VERSIONES>rmdir mi entorno virtual/s
```

mi_entorno_virtual, ¿Está seguro (S/N)?

Marcar S