

Anexo A

Guía para la Configuración de un Entorno de Desarrollo en Python

1. Introducción

La presente guía describe de manera sistematizada los pasos requeridos para la instalación y configuración de un entorno de desarrollo en Python sobre Microsoft Windows, la creación de entornos virtuales, la gestión de dependencias mediante requirements.txt y la elaboración de un instalador ejecutable utilizando la herramienta PyInstaller. El objetivo es proporcionar procedimientos estandarizados y verificables que faciliten la replicabilidad del entorno en futuros proyectos de investigación o desarrollo.

2. Instalación de Python en Windows

2.1 Acceder al sitio oficial: <https://www.python.org/>

2.2 Descargar el instalador correspondiente al sistema operativo.

2.3 Durante el proceso de instalación:

- Seleccionar la opción **“Add Python to PATH”**.
- Elegir **“Customize installation”** y activar **pip** (gestor de paquetes).

2.4 Verificar la instalación mediante la terminal de comandos (cmd):

- `python --version`

- pip --version

3. Creación de un Entorno Virtual

3.1 Abrir la terminal en el directorio raíz del proyecto.

3.2 Ejecutar el comando para crear el entorno virtual:

- python -m venv venv

3.3 Activar el entorno virtual con el siguiente comando:

- En Windows: **.\venv\Scripts\activate**

3.4 Verificar que el entorno esté activo:

- El prompt mostrará el prefijo del entorno, por ejemplo, **(venv) C:\ruta\al\proyecto >**

4. Instalación de las Dependencias

4.1 Asegurarse de tener un archivo **requirements.txt** en la carpeta raíz del proyecto.

4.2 Instalar las dependencias: **pip install -r requirements.txt**

5. Uso de XAMPP para Configuración de Bases de Datos

5.1 Descargar **XAMPP** desde: <https://www.apachefriends.org/>

5.2 Instalar y ejecutar XAMPP.

5.3 En el Panel de Control de XAMPP, iniciar los servicios de **MySQL** y **Apache**.

5.4 Crear la base de datos en phpMyAdmin: <http://localhost/phpmyadmin/>

6. Creación de un Instalador para la Aplicación

Esta sección detalla cómo empaquetar los archivos requeridos (por ejemplo, inventario.sql, config.json, app.py) en un solo instalador utilizando **PyInstaller**.

6.1 Instalación de PyInstaller

6.1.1 Con el entorno virtual activo, instalar PyInstaller: **pip install pyinstaller**

6.2 Empaquetado de la Aplicación

6.2.1 Ubicar los archivos principales del proyecto (app.py, config.json, inventario.sql, etc.) en la carpeta principal del proyecto.

6.2.2 Ejecutar el siguiente comando para crear un ejecutable:

```
pyinstaller --onefile --add-data "config.json;." --add-data "inventario.sql;." app.py
```

6.2.3 Explicación del comando:

- **--onefile**: Genera un solo archivo ejecutable.
- **--add-data**: Añade archivos adicionales al ejecutable.
- **Config.json; . :** Copia config.json al mismo directorio que el ejecutable.

6.3 Ejecución del Instalador

6.3.1 Una vez ejecutado PyInstaller, los archivos generados estarán en la carpeta **dist**.

6.3.2 Para realizar pruebas, ejecutar el archivo **app.exe** en esa carpeta.

6.3.3 Para distribuir la aplicación, comprimir la carpeta **dist** en un archivo ZIP o crear un instalador con herramientas como **Inno Setup**.

7. Conclusión

La guía presentada permite la creación controlada de un entorno de desarrollo en Python, la instalación confiable de dependencias específicas y el empaquetado de aplicaciones en ejecutables listos para distribución. Se recomienda realizar revisiones periódicas de versiones y dependencias para garantizar la vigencia del entorno de desarrollo.

8. Referencias

Iván Ruiz. (2024, 13 de octubre). Guía para la configuración de un entorno de desarrollo en Python [PDF].