

# Requerimientos Tecnicos

---

## Asegurate de tener instalado python

La guía usa como comando `python` o `python3`, en general se usa `python` por default sin embargo puede ser que tu sistema use `python3`, la recomendación es que para conocer si tu sistema operativo tiene `python` pruebes con ambos comandos.

Los comandos son **genericos** en el sentido que tienen **placeholders** para sus los paths y usuarios, asegurate de adecuarlos para tu computadora en específico.

Para instalar Python en diferentes sistemas operativos, sigue esta guía basada en procedimientos verificados:

---

## macOS

### Revisar instalación existente

- Abre **Terminal** y ejecuta:

```
python3 --version # Para Python 3
python --version  # Para Python 2 (si existe)
```

Si no está instalado, verás un mensaje como `command not found`.

### Instalación

#### 1. Método oficial:

- Descarga el instalador desde [python.org/downloads](https://python.org/downloads)
- Ejecuta el archivo `.pkg` y sigue los pasos del asistente

#### 2. Homebrew (opcional):

```
brew install python # Instala Python 3
```

---

## Linux

### Revisar instalación existente

- Ejecuta en la terminal:

```
python3 --version # Versión de Python 3
which python3     # Ubicación del ejecutable
```

## Instalación

- Usa el gestor de paquetes de tu distribución:

```
sudo apt update && sudo apt install python3 # Debian/Ubuntu
```

Para otras distribuciones, reemplaza **apt** con **dnf** (Fedora) o **pacman** (Arch).

---

## Windows (sin WSL2)

### Revisar instalación existente

- Abre **CMD** o **PowerShell** y ejecuta:

```
python --version
py --version # Alternativa si hay múltiples versiones
```

## Instalación

1. Descarga el instalador desde [python.org/downloads](https://python.org/downloads).
  2. Ejecuta el **.exe** y asegúrate de marcar **Add Python to PATH** durante la instalación
- 

## Windows con WSL2

### Revisar instalación existente

- Abre la terminal de WSL (ej. Ubuntu) y ejecuta:

```
python --version # Python en WSL es independiente de Windows
```

## Instalación

1. Actualiza los repositorios e instala Python:

```
sudo apt update && sudo apt install python # Instala Python 3
```

2. Para instalar **pip** (gestor de paquetes):

```
sudo apt install python-pip
```

---

## Verificación post-instalación

En todos los sistemas:

```
python3 --version # Debe mostrar la versión instalada
python3 -c "print('¡Hola, Mundo!')" # Ejecuta un comando rápido
```

Para actualizar Python, repite los pasos de instalación con la última versión disponible. Si usas WSL2, recuerda que su entorno es independiente de Windows.

## Github

Aquí tienes una guía concisa para GitHub, incluyendo verificación de instalación, configuración y vinculación con tu entorno local:

---

## Verificar instalación de Git

Todos los sistemas

```
git --version # Si muestra la versión, Git está instalado
```

---

## Windows (sin GUI)

PROF

```
git version      # En CMD/PowerShell
where git        # Muestra ubicación del ejecutable
```

## Windows con GitHub Desktop

- El CLI de Git **no** se instala globalmente por defecto
- Ubicación alternativa:  
C:\Users\$\$Usuario\AppData\Local\GitHubDesktop\app-[versión]\resources\app\git\cmd

---

## Instalar Git

## Windows

1. Descargar instalador oficial:  
[git-scm.com/downloads](https://git-scm.com/downloads)
2. Marcar **"Add to PATH"** durante la instalación

## macOS

```
# Opción 1: Xcode Tools
xcode-select --install

# Opción 2: Homebrew
brew install git
```

## Linux (Debian/Ubuntu)

```
sudo apt update && sudo apt install git
```

---

## Crear cuenta GitHub

1. Visitar [github.com/join](https://github.com/join)
2. Ingresar:
  - Nombre de usuario único
  - Email válido
  - Contraseña segura
3. Verificar email (buscar en spam/correo no deseado)
4. Opcional: Configurar 2FA

---

PROF

---

## Vincular Git local con GitHub

1. Configurar identidad:

```
git config --global user.name "TuUsuarioGitHub"
git config --global user.email "tu@email.com"
```

2. Autenticación:

```
gh auth login # Usando GitHub CLI
```

o generar SSH key:

```
ssh-keygen -t ed25519 -C "tu@email.com"
cat ~/.ssh/id_ed25519.pub | clip # Copiar clave pública
```

3. Agregar clave SSH en GitHub:

[github.com/settings/keys](https://github.com/settings/keys)

---

## GitHub Desktop (Opcional)

- Descargar desde [desktop.github.com](https://desktop.github.com)
- Se integra automáticamente con Git instalado localmente
- Para usar Git CLI con Desktop: Agregar ruta de instalación al PATH

Verifica la configuración con:

```
git remote -v # Debe mostrar tus repositorios vinculados
```

## Descargar Repo

Para descargar el repositorio de la clase "graph\_zombie\_repo", sigue estos métodos verificados:

---

### Métodos de descarga

#### 1. Vía HTTPS (recomendado para principiantes)

```
git clone https://github.com/sonder-art/graph_zombie_repo.git
```

- **Ventaja:** No requiere configuración SSH
- **Requisito:** Credenciales GitHub válidas si haces push

#### 2. Usando SSH (para usuarios avanzados)

```
git clone git@github.com:sonder-art/graph_zombie_repo.git[1]

ls -la # Debes ver directorios como docs/ y README.md
git branch # Confirma que estás en la rama main/master
```

---

## Solución de problemas comunes

Error: "Repository not found"

- **Causas:**
  - URL mal escrita
  - Permisos insuficientes
- **Solución:**

```
git remote -v # Verifica URL remota
git remote set-url origin https://github.com/sonder-art/graph_zombie_repo.git
```

## Problemas de autenticación HTTPS

```
# Restablece credenciales almacenadas
git config --global --unset credential.helper
git clone https://github.com/sonder-art/graph_zombie_repo.git # Pedirá
usuario/contraseña
```

## Fallos en conexión SSH

```
ssh -T git@github.com # Prueba conexión SSH
# Si falla:
ssh-keygen -t ed25519 -C "tu@email.com" # Genera nueva clave
eval "$(ssh-agent -s)" && ssh-add ~/.ssh/id_ed25519
```

## ZIP corrupto o incompleto

1. Verifica el hash SHA-256 del archivo:

```
shasum -a 256 graph_zombie_repo-main.zip
```

2. Compara con el valor mostrado en **Releases** del repositorio

---

## Notas importantes

- Si usas Windows: Asegúrate de tener Git Bash instalado para comandos UNIX
- Para actualizar el repositorio local:

```
git pull origin main # Sincroniza cambios recientes
```

- Si el repositorio tiene submodules:

```
git submodule update --init --recursive
```

## VSCode (Visual Studio Code)

---

### Verificar instalación existente

#### Windows

```
code --version # Si muestra la versión (ej: 1.85.1), está instalado
where code     # Muestra ruta de instalación (ej: C:\...\Microsoft VS
Code\bin)
```

#### macOS

```
code --version # Versión en terminal
# Alternativa gráfica:
⌘ + Espacio > Buscar "Visual Studio Code" > Abrir
```

#### Linux (incluyendo WSL2)

```
code --version          # Versión CLI
which code              # Ruta de instalación (ej: /snap/bin/code)
ls /usr/share/applications | grep code # Verifica ícono gráfico
```

---

PROF

---

### Instalación por sistema operativo

#### Windows

1. Descarga oficial: [code.visualstudio.com/download](https://code.visualstudio.com/download)
2. Ejecuta `VSCodeUserSetup-x64-*.exe`
3. Marca **Add to PATH** durante la instalación

#### macOS

```
# Opción 1: Descarga directa
Descarga .dmg desde [code.visualstudio.com]
(https://code.visualstudio.com) > Arrastra a Aplicaciones
```

```
# Opción 2: Homebrew
brew install --cask visual-studio-code
```

## Linux/WSL2

```
# Debian/Ubuntu
sudo apt install code # Usa repositorio oficial

# Snap (universal)
sudo snap install code --classic
```

---

## Extensiones esenciales (Python/Jupyter)

Extensión	Enlace	Función
Python	<a href="https://marketplace.visualstudio.com/items?itemName=ms-python.python">marketplace.visualstudio.com/items?itemName=ms-python.python</a>	Autocompletado, depuración y entornos virtuales
Jupyter	<a href="https://marketplace.visualstudio.com/items?itemName=ms-toolsai.jupyter">marketplace.visualstudio.com/items?itemName=ms-toolsai.jupyter</a>	Ejecución de notebooks .ipynb
GitHub Copilot	<a href="https://marketplace.visualstudio.com/items?itemName=GitHub.copilot">marketplace.visualstudio.com/items?itemName=GitHub.copilot</a>	IA para autocompletado de código
GitLens	<a href="https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens">marketplace.visualstudio.com/items?itemName=eamodio.gitlens</a>	Historial de cambios y colaboración

**Instalar extensiones:** `Ctrl/Cmd + Shift + X` > Buscar nombre > Install

---

## Troubleshooting básico

PROF

### Comando 'code' no reconocido

- **Windows:** Reinstalar marcando **Add to PATH**
- **macOS/Linux:** Ejecutar desde la app VS Code: `⌘ + ⌘ + P` > `Shell Command: Install PATH`

### Problemas con WSL2

```
# En VS Code local:
Instalar extensión "Remote - WSL"
Presionar `F1` > "Remote-WSL: New Window"
```

## Verificar versión ARM (Apple Silicon)



1. Finder > Aplicaciones > Click derecho en VS Code > "Obtener información"
  2. En **Tipo**: Debe decir "Aplicación (Apple silicon)"
- 

### Primeros pasos recomendados

1. Abrir terminal integrada: **Ctrl + Ñ**
2. Crear nuevo Jupyter Notebook: **Ctrl + N** > Seleccionar ".ipynb"
3. Conectar a GitHub: **Ctrl + Shift + G** > Iniciar sesión