

# Guía práctica: Uso de Python, VS Code y Git en computador personal

---

**Docente:** Lina María Castro

**Asignatura:** Inteligencia Artificial con Aplicaciones en Economía I  
Universidad Externado de Colombia

## Objetivos de aprendizaje

- Instalar Python, VS Code y Git.
- Hacer fork y clonar un repositorio de Github.
- Crear y activar un ambiente virtual.
- Conocer Jupyter Notebook e instalar librerías.
- Iniciar un proyecto con control de versiones en Git y publicarlo en GitHub.

## 1. Introducción a Python y su utilidad en economía

Python es un lenguaje de programación versátil, gratuito y de código abierto. En economía, es ampliamente utilizado para análisis de datos, modelos econométricos, scraping de datos públicos y visualización de resultados. Algunas bibliotecas clave incluyen pandas, numpy, matplotlib, statsmodels y scikit-learn.

## 2. Instalación del entorno de trabajo

Sigue estos pasos para preparar tu entorno de desarrollo:

### a. Instalar Python

- Ir a <https://www.python.org/downloads/>
- Descargar la última versión estable para el sistema operativo del computador.
- Abrir el ejecutable.
- Marcar "Add Python to PATH" y seleccionar "Install Now".
- Si se quiere seleccionar una carpeta diferente en la instalación, cambiarle el nombre a la carpeta donde va a quedar el programa o cambiar algo en la configuración, en lugar de seleccionar "Install Now", seleccionar "Customize installation" y hacer los cambios necesarios.

## b. Instalar Visual Studio Code (VS Code)

- Ir a <https://code.visualstudio.com/download>
- Descargar según sistema operativo del computador.
- Abrir el ejecutable e instalar.
- Abrir el programa para verificar que haya quedado instalado.
- **Verificar que Python esté en el PATH:**
  - o Abrir terminal de VS Code.
  - o Escribir en la terminal: `python --version`.
  - o Si no sale un número de versión y dice que no se reconoce el comando, entonces Python no quedó en el PATH y hay que reinstalarlo marcando la opción "Add Python to PATH".
  - o Si sale la versión de Python, entonces quedó bien la instalación.
- **Activar la extensión de Python:**
  - o Ir al icono de Extensiones en el menú de la izquierda.
  - o Buscar "Python" e instalar la extensión que publica Microsoft.
  - o Reiniciar VS Code.
- **Activar la extensión de Jupyter Notebook:**
  - o Ir al icono de Extensiones en el menú de la izquierda.
  - o Buscar "Jupyter" e instalar la extensión que publica Microsoft.
  - o Reiniciar VS Code.
- **Instalar Jupyter ipykernel:**
  - o Abrir la terminal de VS Code.
  - o Escribir: `pip install jupyter ipykernel`

## c. Instalar Git

- Ir a <https://git-scm.com/downloads>
- Descargar según sistema operativo del computador.
- Abrir el ejecutable e instalar seleccionando lo siguiente:
  - o En la pantalla de opciones, dejar las opciones por defecto, pero asegurarse que "Git Bash Here" esté activado.
  - o Seleccionar Visual Studio Code como editor por defecto.
  - o Seleccionar "Override the default branch name for new repositories" y verificar que diga "main".
  - o Seleccionar: "Git from the command line and also from 3rd-party software"
  - o Seleccionar: "Checkout Windows-style, commit Unix-style line endings"
  - o Seleccionar: "Use MinTTY (the default terminal of MSYS2)"
  - o Seleccionar: Fast-forward or merge
  - o Seleccionar: Do not use a credential helper
  - o Seleccionar: "Enable file system caching"
- Abrir una cuenta en Github: <https://github.com/>
- Abrir la terminal de VS Code.

- Verificar que tenga instalado Git y cuál es la versión, escribiendo en la terminal: `git --version`
- Cambiar o agregar el nombre de usuario: `git config --global user.name "TuUsuarioDeGithub"`
- Utilizar el correo asociado a Github: `git config --global user.email "tuemaildegithub@xxxx.com"`
- Verificar que se haya registrado correctamente el usuario: `git config user.name`
- Verificar que se haya registrado correctamente el email: `git config user.email`
- Configurar la rama main como rama principal en lugar de master: `git config --global init.defaultBranch main`
- Mejorar la visualización en la terminal: `git config --global color.ui true`

### 3. Hacer Fork de un repositorio en Github

**Fork:** Copia un repositorio en tu cuenta de GitHub para trabajar independientemente o contribuir. Los cambios que hagas en tu fork no afectan al repositorio original. Puedes hacer commits, crear ramas y modificar el código sin preocuparte por alterar el proyecto original. Si hiciste un fork y el repositorio original se actualiza, puedes sincronizar tu fork para incorporar los cambios. Si quieres contribuir a un proyecto, haces un fork, realizas tus cambios y luego envías un pull request al repositorio original para que el mantenedor revise y acepte tus cambios.

En este caso particular, vas a realizar un fork del repositorio del curso. Pasos:

- Iniciar sesión en Github.
- Ir al repositorio en GitHub que se desea copiar, en este caso: <https://github.com/LinaMariaCastro/curso-ia-para-economia.git>
- En la esquina superior derecha del repositorio, hay un botón que dice Fork. Hacer clic en él.
- Seleccionar tu cuenta. Si tienes varias cuentas o perteneces a organizaciones, GitHub te pedirá que selecciones dónde quieres hacer el fork.
- El repositorio se copiará en tu cuenta de Github.

### 4. Clonar un repositorio que está en Github

**Clone:** Descarga un repositorio que está en Github en tu computador para trabajar localmente y sincronizar cambios con Github.

En este caso particular, vas a clonar el repositorio del curso que tienes en tu cuenta personal de Github. Pasos:

- Crear una carpeta en el computador donde vas a ubicar el curso.
- En Visual Studio Code, ir a Open Folder y ubicarse en la carpeta del curso.

- Ir al repositorio en Github, hacer clic en el botón verde "Code" y copiar el https que aparece ahí.
- Abrir una nueva terminal en VS Code.
- Escribir: git clone <https://github.com/TuUsuarioEnGithub/nombre-del-repositorio.git>

## 5. Crear un ambiente virtual e instalar requirements

- Abrir VS Code, ir a Open Folder y ubicarse en la carpeta que creaste para el curso.
- Ir a File, New file, Jupyter Notebook.
- Ir a Select Kernel, Python environment, create Python environment, venv.
- Seleccionar la version de Python deseada (anotar cuál se usó).
- Seleccionar el archivo requirements.txt.
- Click en "show logs" para ver el progreso de la creación del ambiente virtual. Cuando se detenga ya quedó listo (al final usualmente dice "commandPrompt").
- Cuando se abra un Jupyter Notebook, seleccionar en el Kernel el ambiente virtual creado.

## 6. Jupyter Notebook y uso de librerías

- Abrir un Jupyter Notebook. Ir a File, New file, Jupyter Notebook para empezar desde cero o dar clic sobre un notebook que ya exista para abrirlo.
- Siempre lo primero, es importar las librerías que se requieren para el código.
- Luego, escribir el código deseado en cada celda.
- Para ejecutar el código se da clic en el triángulo o con Ctrl+Alt+Enter. El resultado sale inmediatamente debajo de la celda.
- En los notebooks se pueden incluir celdas de Markdown para colocar títulos o comentarios.

## 7. Subir cambios en el código local a Github

- Abrir la terminal de VS Code.
- Para añadir todos los archivos de la carpeta y sus cambios, escribir: git add .
- Opcional: si se quiere saber todo lo que se añadió al repositorio y ver qué archivos no han sido registrados, escribir en la terminal: git status -s
- Para crear una fotografía del proyecto en ese momento y colocarle una descripción de lo que contiene, escribir: git commit -m "descripciondeloscambios"
- Para enviar los cambios a Github, escribir: git push

## 8. Descargar cambios en el código en GitHub al equipo local

- Abrir la terminal de VS Code y escribir: git pull origin main

## 9. Recibir actualizaciones de un repositorio al cual se le hizo Fork

Como se hizo un Fork del repositorio del curso, es importante recibir las actualizaciones que se hagan en ese repositorio para obtener la información de cada clase. Pasos:

- Entrar a Github y a tu fork del repositorio del curso.
- Dar clic en el botón "Sync fork" y luego en "Update branch". Esto actualiza el fork directamente en GitHub, pero no actualiza la copia local en el computador.
- Para actualizar el repositorio local, ir a la terminal de VS Code y escribir: `git pull origin main`

## 10. Crear un repositorio en GitHub para el proyecto del curso

Como el proyecto es en equipos, escoger un líder que cree el repositorio en Github y los otros compañeros clonan el repositorio. El líder debe invitar a sus dos compañeros como colaboradores del repositorio, así todos pueden contribuir en él.

Dejar el repositorio como público para que las demás personas lo podamos ver (profesores y compañeros).

El líder debe hacer lo siguiente:

- Crear en su computador una carpeta para el proyecto.
- En Visual Studio Code, ubicarse en la carpeta del proyecto.
- Crear un ambiente virtual e instalar las librerías necesarias para el proyecto.
- Abrir una terminal de VS Code.
- Para ver todo lo instalado en el ambiente virtual, se puede escribir en la terminal de VS Code: `pip freeze`
- Crear el archivo de requerimientos del proyecto con todas las librerías necesarias a instalar. Escribir en la terminal: `pip freeze > requirements.txt`
- Si posteriormente se instalan más librerías, se debe hacer nuevamente este paso para actualizar el archivo de requirements.txt y subir la última versión a Github.
- Crear un nuevo repositorio (crea la carpeta oculta .git, solo se ejecuta una vez por proyecto) escribiendo: `git init`
- Escribir: `git add .`
- Escribir: `git commit -m "inicio del repositorio"`
- Ir a la cuenta en Github y en el + escoger "create new repository", darle un nombre al repositorio (usualmente se hace en minúscula y se coloca guión en lugar de espacios), seleccionar la opción para que el repositorio sea público y dar clic en "crear". No dar clic en crear README, este se creará después.
- En el repositorio en Github, copiar el https que aparece ahí.
- En la terminal de VS Code, indicar la ruta del repositorio en Github escribiendo: `git remote add origin https://github.com/TuUsuarioEnGithub/nombre-del-repositorio.git`
- Indicar la rama donde va a quedar la información: `git branch -M main`

- Enviar la información al repositorio en Github: `git push -u origin main`
- Sale un cuadro de diálogo donde se debe colocar las credenciales de la cuenta en Github.
- Invitar a los otros miembros del equipo como colaboradores:
  - o Ir a Settings > Collaborators.
  - o Clic en Invite a collaborator.
  - o Escribir los nombres de usuario o email de los otros compañeros de equipo.
  - o Enviar la invitación.
- Informar a sus compañeros de equipo, la versión de Python que deben tener para el proyecto.

Los otros compañeros de equipo deben:

- Aceptar la invitación.
- Clonar el repositorio en su computador ([recordar instrucciones](#)).
- Crear el ambiente virtual e instalar los requirements ([recordar instrucciones](#)).

Luego de esto, todos pueden trabajar localmente en sus computadores personales y hacer push/pull de los cambios. Sin embargo, hay que tener cuidado al hacer push de no enviar algo que pueda dañar el código que está en Github o al hacer pull de no traer algo al computador local que dañe el código propio. Para evitar esto, es recomendable dividirse el trabajo, que cada persona trabaje en partes diferentes (no tener 2 personas haciendo lo mismo) y coordinarse muy bien entre ustedes para que todo el proyecto sea congruente y se pueda unificar.