

---

# Guía Completa de Instalación: Entorno de Estadística Aplicada y Agentes de IA

El objetivo es configurar el entorno de desarrollo **Visual Studio Code (VSC)** con soporte para **Python** y **R** (usando **Jupyter Notebooks**), integrar **GitHub Copilot** como asistente de código, y entender la **Orquestación de Agentes con el protocolo MCP**.

---

## 1. Descarga e Instalación de Software Base

Componente	Enlace de Descarga/Sitio Web Oficial	Notas Cruciales de Instalación
Python 	<a href="https://www.python.org/downloads/">https://www.python.org/downloads/</a>	<b>¡CRUCIAL!</b> En la primera pantalla, <b>MARCAR</b> la casilla <b>"Add python.exe to PATH"</b> .
R 	<a href="https://cran.r-project.org/">https://cran.r-project.org/</a>	Descargar la versión para <b>Windows</b> (generalmente la opción <i>base</i> ).
RStudio 	<a href="https://posit.co/download/rstudio-desktop/">https://posit.co/download/rstudio-desktop/</a>	Descargar la versión <b>Desktop (Open Source License)</b> .
Visual Studio Code (VSC) 	<a href="https://code.visualstudio.com/download">https://code.visualstudio.com/download</a>	Descargar el instalador de Windows.
Git para Windows (Requisito para GitHub)	<a href="https://git-scm.com/download/win">https://git-scm.com/download/win</a>	Necesario para que VSC se conecte a GitHub y use Copilot.

- VSC actúa como un cliente Git:** Cuando haces un **clone, commit, push, pull, merge**, etc., desde VSC, en realidad VSC está ejecutando comandos de Git en segundo plano. Para que pueda ejecutar esos comandos, necesita que la herramienta **git.exe** esté instalada y accesible en tu **PATH** de Windows.
- Repositorios locales:** Git es un sistema de control de versiones distribuido. Incluso cuando trabajas con GitHub (que es un servicio de hosting de repositorios Git), estás interactuando con una copia local de tu repositorio Git en tu máquina. **git.exe** es lo que crea, gestiona y sincroniza esa copia local.

3. **Configuración de Git:** Comandos como `git config --global user.name` y `git config --global user.email` (que VSC puede usar o solicitarte configurar) son comandos de Git que necesitan la instalación base de Git.

En resumen:

- **Sin Git instalado:** VSC no podrá clonar repositorios, hacer `commits`, `pushes`, `pulls` ni ninguna otra operación de control de versiones con GitHub.
- **Con Git instalado:** VSC detectará la instalación de Git y te permitirá realizar todas las operaciones de GitHub de forma visual y sencilla desde su interfaz.

---

## 2. Configuración de PATH y Kernels

### A. PASO CRUCIAL: Añadir R al PATH de Windows

Para que R pueda ser llamado por Jupyter y otras herramientas del sistema, debes añadir su ubicación al **PATH** de Windows:

1. **Encuentra la Ruta de R:** La ruta de instalación de R es típicamente: `C:\Program Files\R\R-x.x.x\bin\x64` (reemplaza `R-x.x.x` con tu versión instalada).
2. **Edita Variables de Entorno:**
  - Busca en el menú de inicio de Windows: **"Editar las variables de entorno del sistema"**.
  - En la ventana que se abre, haz clic en el botón **"Variables de entorno..."**.
  - En la sección **"Variables del sistema"**, busca y selecciona **Path** y haz clic en **"Editar"**.
  - Haz clic en **"Nuevo"** y pega la ruta que encontraste en el paso 1 (ej: `C:\Program Files\R\R-4.4.1\bin\x64`).
  - Haz clic en **Aceptar** para guardar los cambios.

### B. Instalación de Extensiones de VSC

Abrir **Visual Studio Code** y usar el menú de **Extensiones** (Ctrl+Shift+X) para instalar:

Extensión	ID de la Extensión (Para búsqueda)	Función Principal
<b>Python</b>	<code>ms-python.python</code>	Soporte principal para el lenguaje Python.
<b>Jupyter</b>	<code>ms-toolsai.jupyter</code>	Habilita la creación y ejecución de Notebooks.

<b>R</b>	<code>REditorSupport.r</code>	Soporte para sintaxis y <i>snippets</i> de R.
<b>GitHub Copilot</b>	<code>GitHub.copilot</code>	El asistente de codificación de IA.

## C. Configuración de Jupyter (Python y R)

### 1. Instalar el paquete principal de Jupyter (usando Python):

Abrir el **Símbolo del sistema (CMD)** o PowerShell y ejecutar:

```
Bash  
pip install jupyter
```

### 2. Instalar y Registrar el Kernel de R (IRkernel):

Abrir la **Consola de R** (o la consola de RStudio) y ejecutar los siguientes comandos:

```
R  
1. Instala el paquete IRkernel  
install.packages("IRkernel")  
  
2. Registra el kernel de R en Jupyter  
IRkernel::installspec()
```

---

## 3. Activación de GitHub Copilot

1. **Iniciar Sesión en GitHub en VSC:** VSC pedirá iniciar sesión y **autorizar VSC** a acceder a la cuenta de GitHub.
2. **Verificar Suscripción:** **GitHub Copilot** requiere una **suscripción activa** asociada a la cuenta de GitHub que se realiza teniendo una cuenta en gmail.
3. **Verificación Final:** El icono de Copilot se iluminará en la barra de estado inferior de VSC, indicando que está listo.

---

## 4. Agentes de IA, Orquestación MCP

### Orquestador de Agentes en VSC (Protocolo MCP)

El "Orquestador" es el "Agent Mode" de VSC (parte de Copilot Chat), que utiliza el protocolo **MCP (Model Context Protocol)** para coordinar agentes especializados llamados **MCP Servers**.

Rol	Definición en VSC
<b>Orquestador / Cliente</b>	<b>Visual Studio Code (Agent Mode de Copilot Chat).</b>
<b>Protocolo</b>	<b>Model Context Protocol (MCP):</b> Estándar para que el agente interactúe con herramientas externas de forma segura.
<b>Herramientas (Servidores)</b>	<b>MCP Servers</b> (Ej: <a href="#">GitHub MCP Server</a> ). Son la lógica especializada para interactuar con sistemas externos (APIs, bases de datos). Para tener acceso a cuentas externas.

### Procedimiento para Instalar un MCP Server:

1. Abre la paleta de comandos de VSC (Ctrl+Shift+P).
2. Ejecuta el comando: **MCP: Browse Servers** o **MCP: Add Server**.
3. Selecciona e instala el servidor deseado.
4. En el **Agent Mode** de VSC, el agente de IA usará automáticamente estas herramientas para responder a tus solicitudes, pidiendo **aprobación** para cualquier acción crítica.