

Guía de instalación paso a paso

Todo lo necesario para los ejercicios: VS Code, Julia, Python y RAMP

Dra. Tatiana González Grandón
Norwegian University of Science and Technology

Haga esta instalación antes de la semana del curso. Son unos 30 minutos de pasos más las descargas (unos 3 GB en total); el último paso deja los paquetes ya descargados para que en clase todo arranque en segundos. No necesita saber programar: basta con seguir cada paso en orden.

1. Qué va a instalar

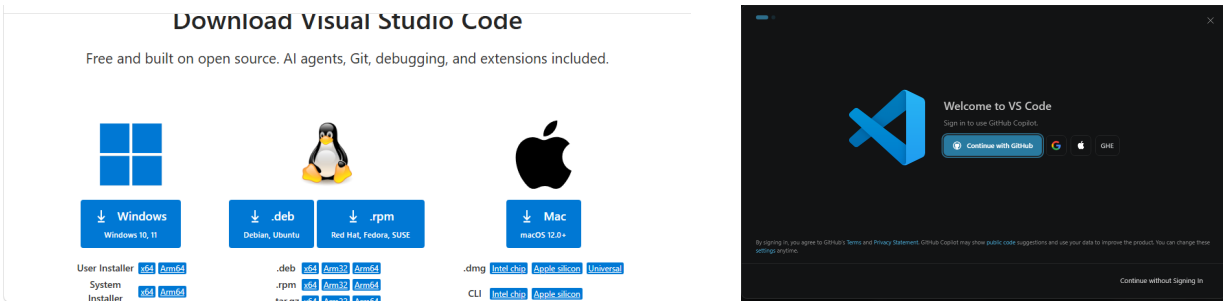
Herramienta	Para qué	Paso
VS Code	el editor donde se abre y ejecuta todo	Sección 2
Extensiones de Julia y de Python	el botón de Play para cada lenguaje	Sección 3
Julia (1.12)	los modelos de optimización (JuMP + HiGHS)	Sección 4
Python 3.11 y RAMP	la demanda eléctrica del pueblo	Sección 5
Paquetes de Julia	se instalan solos la primera vez	Sección 6

Al final hay una lista de comprobación (Sección 7). Los comandos se escriben en la *terminal*: en VS Code, *Terminal* → *New Terminal*.

2. VS Code

Descargue el instalador de <https://code.visualstudio.com/download> y ejecútelo como cualquier otro programa (basta aceptar la licencia y pulsar *Next* hasta *Install*). Elija su sistema: el botón **Windows** (“Windows 10, 11”), el botón **Mac** (“macOS 12.0+”) o, en Linux, el paquete **.deb** (Debian, Ubuntu) o **.rpm** (Red Hat, Fedora, SUSE).

Al abrir VS Code por primera vez aparece una pantalla de bienvenida que invita a iniciar sesión con GitHub (Copilot): no necesita ninguna cuenta; pulse *Continue without Signing In* (abajo a la derecha) y después *Get Started*.

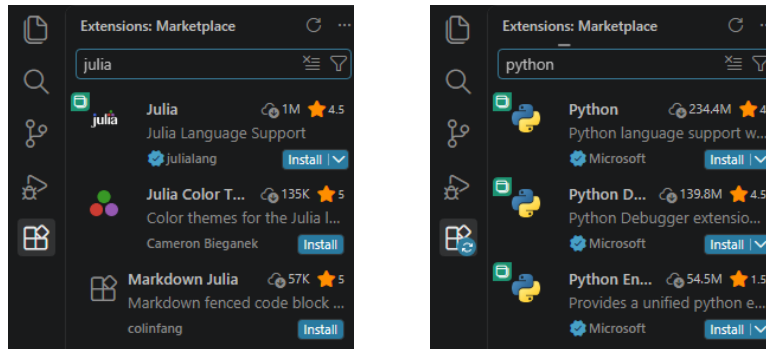


3. Las dos extensiones

Abra VS Code, pulse el icono de *Extensiones* (o **Ctrl+Shift+X**) y busque e instale estas dos, comprobando el editor que aparece bajo cada resultado (figuras):

- **Julia**, del editor **julia**lang (“Julia Language Support”).

- **Python**, del editor **Microsoft**.



Alternativa equivalente: pulse **Ctrl+P** y pegue `install julialang.language-julia` (y después `install ms-python.python`). La primera vez que instale una extensión, VS Code pregunta si confía en el editor: pulse **Trust Publisher & Install**.

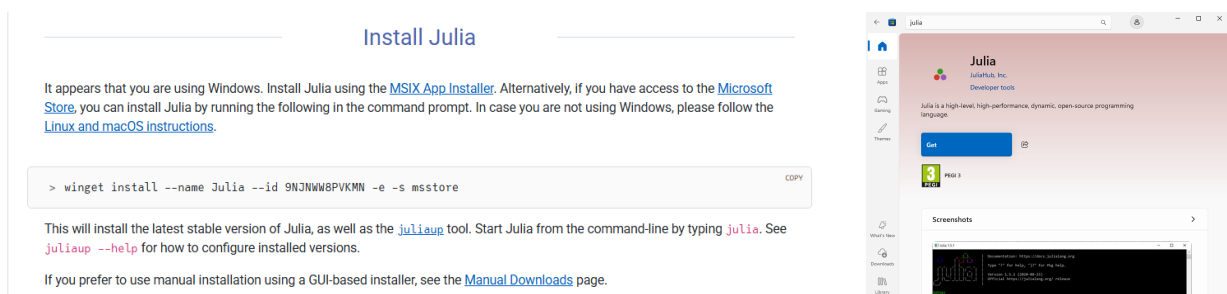
4. Julia

En <https://julialang.org/downloads/> (“Install Julia”):

- **macOS y Linux**: copie el comando de la página en una terminal:

```
curl -fsSL https://install.julialang.org | sh
```
- **Windows**: la página detecta su sistema y ofrece directamente el instalador “MSIX App Installer”. Equivalente: abra la aplicación **Microsoft Store**, busque **Julia** (editor JuliaHub, Inc., gratuita) y pulse **Obtener** (*Get* si su Windows está en inglés, como en la figura).

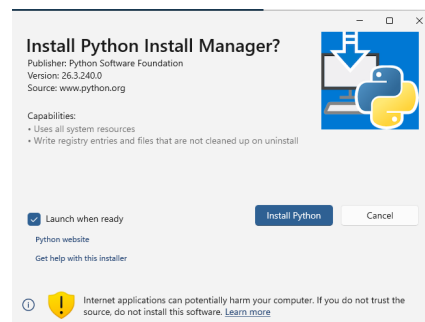
Esto instala *juliaup*, que gestiona las versiones (sirve cualquier Julia 1.12.x). Compruébelo: abra una terminal **nueva**, escriba `julia` y debe aparecer el intérprete; salga con `exit()`. La primera vez tarda algo más: antes del prompt `julia>` descarga la versión actual (verá *Installing Julia 1.12.x*); espere a que termine.



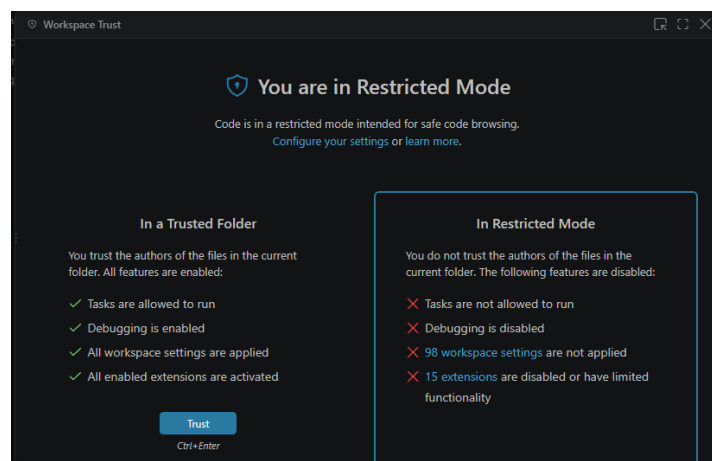
5. Python 3.11 y RAMP

RAMP (<https://rampdemand.org>, el simulador de demanda) necesita **Python 3.11**, pero no hace falta instalar esa versión a mano ni tocar el Python del sistema: su propio Python crea un pequeño entorno desechable, ese entorno instala la herramienta *uv*, y *uv* crea el entorno 3.11 (descargando 3.11 si su ordenador no lo tiene).

Requisito: cualquier Python 3 con el módulo `venv`. Compruébelo escribiendo `python --version` en una terminal. Si no tiene ninguno: en **Windows**, al escribir `python` se abre la Microsoft Store en la página de **Python Install Manager** (de Python Software Foundation); instálelo desde ahí, o desde <https://www.python.org/downloads/> (botón “Download Python install manager”). Tras instalarlo se abre una ventana negra con varias preguntas en inglés: pulse Enter en todas (las respuestas por defecto sirven; una de ellas descarga Python). En macOS descargue Python del enlace “macOS” de esa página; en Debian/Ubuntu, `sudo apt install python3-venv`.



Descomprima el material del ejercicio y abra su carpeta `demand_ramp` en VS Code (*File* → *Open Folder*; pulse **Trust** si VS Code pregunta si confía en la carpeta): la terminal nueva se abre ya **dentro de demand_ramp**.



Ejecute, línea a línea:

macOS / Linux

```
python3 -m venv bootstrap
bootstrap/bin/pip install uv
bootstrap/bin/uv venv --python 3.11 --seed venv
source venv/bin/activate
pip install -r requirements.txt
python ramp_village.py
```

Windows (PowerShell, la terminal por defecto de VS Code)

```
python -m venv bootstrap
bootstrap\Scripts\pip install uv
bootstrap\Scripts\uv venv --python 3.11 --seed venv
venv\Scripts\Activate.ps1
pip install -r requirements.txt
python ramp_village.py
```

La terminal de VS Code en Windows es PowerShell. Si `Activate.ps1` da un error rojo (*execution policy*), ejecute una vez `Set-ExecutionPolicy -Scope CurrentUser RemoteSigned` y repita la activación.

```
Terminal ... powershell + v ~ [ ] [ ] ... | [ ] X
see about_Execution_Policies at
https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ venv\Scripts\Activate.ps1
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Users\homer\Documents\material2_estudiantes\minired_solar\ejercicio\demand_ramp>
```

Solo si usa cmd, la activación es `venv\Scripts\activate.bat` (cada variante funciona solo en su terminal: la de cmd no hace nada en PowerShell). Con el entorno activado, el prompt empieza por `(venv)`. La carpeta `bootstrap` puede borrarse al terminar. La última línea ya es la primera ejecución: debe imprimir una tabla de demanda en unos segundos. Para el botón de Play, pulse la versión de Python en la barra inferior de VS Code (o `Ctrl+Shift+P`, *Python: Select Interpreter*) y elija el intérprete `venv`; a veces VS Code preselecciona `bootstrap` y hay que cambiarlo.

6. Los paquetes de Julia (JuMP y compañía)

Nada que instalar a mano: cada script de Julia del material instala sus paquetes exactos la primera vez que se ejecuta (`Pkg.instantiate`, con las versiones fijadas en `Project.toml` y `Manifest.toml`).

Lo único importante: abra la carpeta `dispatch` del material en VS Code (*File* $\rightarrow$ *Open Folder*; pulse **Trust** si pregunta, y si la extensión de Julia pregunta por telemetría responda lo que prefiera), abra `dispatch_24h.jl` y pulse el **botón de Play** (*Julia: Execute active File in REPL*) **antes de la semana del curso**. La primera vez tarda entre 5 y 20 minutos según el ordenador y descarga ~ 2 GB (verá *Installing artifacts* y *Precompiling packages*); déjelo terminar: al final imprime una tabla de resultados y guarda las figuras en la carpeta `output`. Las siguientes ejecuciones tardan segundos.

7. Lista de comprobación

- VS Code abre.
- Extensiones **Julia** (julialang) y **Python** (Microsoft) instaladas.
- **julia** en una terminal abre el intérprete.
- `python ramp_village.py` imprime la tabla de demanda (con el **venv** activado).
- Un script de Julia ejecutado una vez con Play, hasta el final.

Si algo falla, traiga el mensaje de error a clase: se resuelve en un minuto y no bloquea al resto del grupo.