

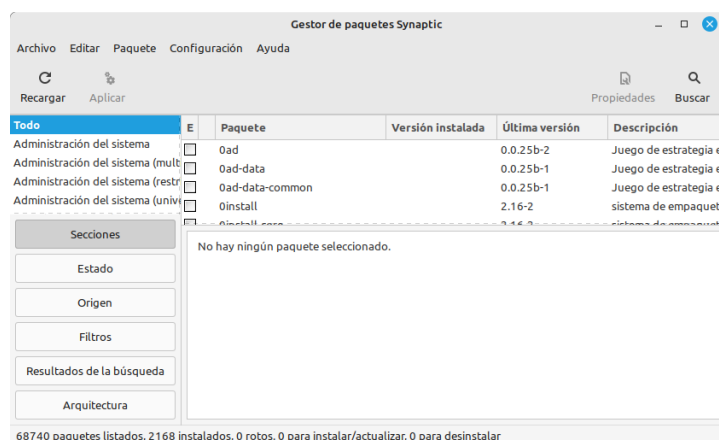
Sublime Text para Python en Linux Mint

Sublime es un editor de texto de código libre que admite numerosas modificaciones. En este tutorial transformaremos Sublime en un ligero y eficaz IDE con el que:

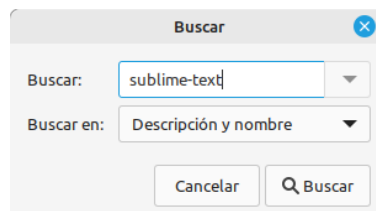
- Programar en Python utilizando un Linux (Mint) como sistema operativo.
- Utilizar la consola integrada Terminus
- Utilizar funciones de autocompletado, análisis de errores y estilo de código.

1. Descarga e instalación

Abre el **gestor de paquetes synaptic**. Tras identificarte con tus credenciales de administrador se abrirá la ventana principal:



Haz clic en el **botón Buscar** y escribe la expresión **sublime-text**:



Se mostrará en la ventana de la derecha el paquete sublime-text. Haz clic sobre su nombre con el clic derecho del ratón y selecciona la opción **marcar para instalar**. Y a continuación haz clic en el **botón Aplicar**. Se abrirá una nueva ventana informando de los detalles de la instalación. Haz clic en el **botón Aplicar**. Al terminar la instalación ya podrás ejecutar SublimeTex.

En el caso de Linux Mint, se habrá creado (si todavía no existía) un grupo nuevo de programas en el **menú de inicio** llamado **programación**.

Desde el terminal puedes ejecutar la siguiente orden para ver que versión de Sublime has instalado:

```
subl --version
```

En este ejemplo:

```
Sublime Text Build 4200
```

2. Preparar el sistema

Versión Python

Puedes comprobar desde un terminal la versión de Python instalada en el equipo:

```
python3 --version
```

Suponiendo que estás en un equipo de nombre pc, utilizando el usuario user:

```
user@pc:~$ python3 --version  
Python 3.12.3
```

Nos indica que la versión instalada de Python es la 3.12.3

Instalación de pip

- pip es el gestor de paquetes de Python, que permite instalar librerías externas como flake8, pylint, autopep8, numpy, pandas, etc.
- Aunque se podrían instalar algunas librerías con apt, las versiones suelen estar desactualizadas y no permiten aislar dependencias por proyecto.

Instala pip desde terminal (Es posible que ya esté instalado):

```
sudo apt install python3-pip
```

Entorno virtual

Un **entorno virtual** es un espacio aislado dentro del sistema donde se instala una copia independiente de Python y sus librerías.

Utilizar un entorno virtual permite que cada proyecto tenga **sus propias versiones** de paquetes sin afectar al resto del sistema ni a otros proyectos.

De forma opcional puedes crear un entorno virtual para tus proyectos

```
python3 -m venv ~/python-env  
source ~/python-env/bin/activate
```

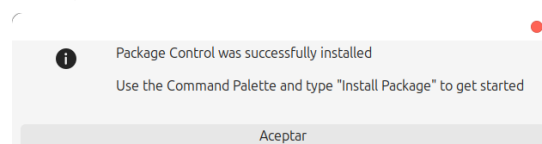
3. Instalar Package Control

Package Control es el gestor oficial de paquetes de Sublime Text.

Permite **buscar, instalar, actualizar y eliminar** extensiones desde el propio editor.

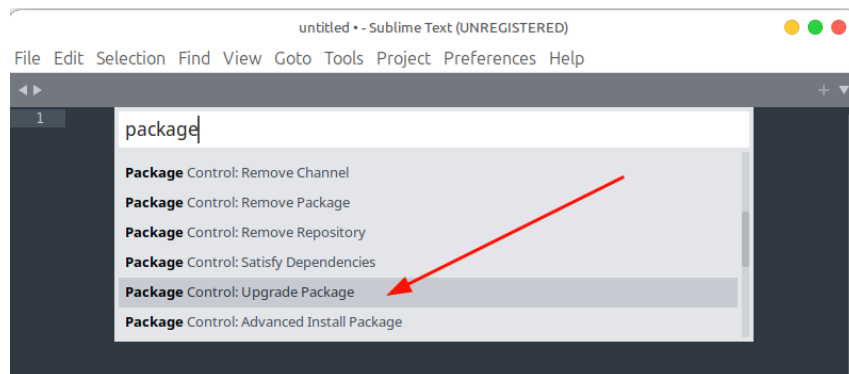
1. **Abre Sublime Text.**
2. Ve a **Tools → Command Palette...** (Ctrl+Shift+P).
3. Escribe **Install Package Control** y pulsa Enter.

Esto instalará el gestor de paquetes de Sublime, que permite buscar, instalar y mantener actualizados componentes mejorando su funcionalidad. Al terminar se mostrará:

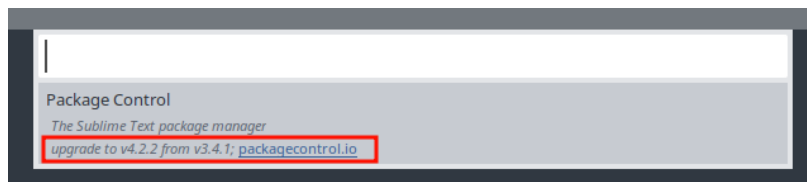


Antes de continuar, actualiza Package Control:

4. Ve a Tools → Command Palette y ejecuta **Package Control: Upgrade Package Control**.



5. Al terminar se mostrará la versión actualizada:



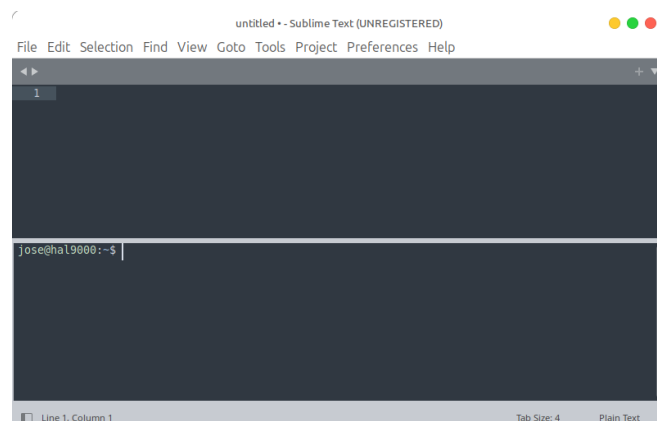
6. Reinicia Sublime Text

4. Instalar Terminus

Terminus es un plugin para Sublime Text que permite abrir y utilizar una terminal directamente dentro del editor.

1. Abre **Tools** → **Command Palette...** (Ctrl+Shift+P) y selecciona **Install Package**.
2. Busca **Terminus** y pulsa Enter para instalarlo.
3. Una vez instalado, abre la terminal integrada con: **Tools** → **Command Palette** → **Terminus: Open Default Shell**.

Se mostrará la terminal en la parte inferior de la pantalla de Sublime



5. Configuración para ejecutar Python con Terminus

Una vez terminada la instalación, crearemos un nuevo Build System que nos permita utilizar Terminus de forma adecuada.

Build system: Conjunto de configuraciones y comandos que permite compilar, ejecutar o realizar otras tareas relacionadas con el desarrollo de software.

Crear un Build System personalizado

1. Ve a Tools → Build System → New Build System...
2. Borra todo el contenido y pega lo siguiente:

```
{
  "target": "terminus_exec",
  "cancel": "terminus_cancel_build",
  "focus": true,
  "timeit": false,

  "cmd": ["python3", "-u", "$file"],
  "file_regex": "^\\[ \\]*File \\(\\.\\.\\.\\?\\)\\", line \\([0-9]\\)*",
  "selector": "source.python",
  "env": {"PYTHONIOENCODING": "utf-8"},

  "windows": {
    "cmd": ["py", "-u", "$file"]
  },

  "variants":
  [
    {
      "name": "Syntax Check",
      "cmd": ["python3", "-m", "py_compile", "$file"],
      "windows": {
        "cmd": ["py", "-m", "py_compile", "$file"]
      }
    }
  ]
}
```

La clave para que input() funcione de forma correcta es "target": "terminus_exec" y "focus": true.

3. Guarda el archivo con un nombre reconocible, por ejemplo: **Python Terminus.sublime-build**

Configurar atajos de teclado (Key Bindings)

Vamos a definir los siguientes atajos (puedes modificar las combinaciones de teclas).

- Ctrl+Alt+B → Ejecuta tu script con el Build System de Terminus
- Alt+T → Abre una terminal Terminus manual
- Alt+, → Cierra la consola de Terminus

Para ello:

1. Ve a Preferences → Package Settings → Terminus → Key Bindings
2. Sustituye el contenido con algo como esto:

```
[
  { "keys": ["ctrl+alt+b"], "command": "build" },
  { "keys": ["alt+t"], "command": "terminus_open",
    "args": { "cmd": "bash", "cwd": "${file_path:${folder}}", "panel_name": "Terminus" }
  },
  { "keys": ["alt+,"], "command": "terminus_close",
    "context": [{ "key": "terminus_view" }] }
]
```

3. Guarda el archivo.

Ejecutar un script Python interactivo

1. Abre un archivo .py cualquiera.
2. Selecciona Tools → Build With... → Python Terminus
3. Pulsa Ctrl+Alt+B para ejecutar el script.

Cualquier input() funcionará correctamente en el panel de Terminus.

Opcional: Estilo de la consola Terminus

1. Ve a Tools → Command Palette → Terminus: Select Theme
2. Elige un tema de colores, por ejemplo: solarized-dark-higher-contrast

Se aplicará automáticamente a la consola de Terminus.

6. Instalar LSP + LSP-Pyright

LSP y LSP-Pyright añaden a Sublime Text **autocompletado inteligente y detección de errores en tiempo real** para Python.

Instalación:

1. Ctrl+Shift+P → Install Package → LSP
2. Ctrl+Shift+P → Install Package → LSP-pyright

Configuración (si usas entorno virtual):

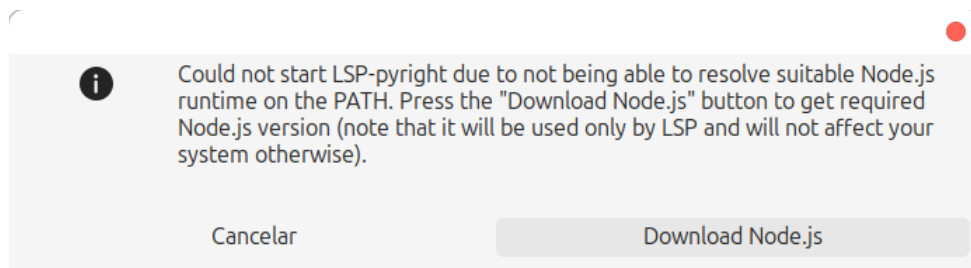
3. Ve a Preferences → Package Settings → LSP → Settings
4. Añade la ruta a tu intérprete Python:

```
"python.pythonPath": "/home/usuario/python-env/bin/python"
```

Donde usuario ha de ser el nombre del usuario que está creando su entorno virtual.

Verificar la configuración de LSP-Pyright

Al abrir un script Python, puede aparecer un mensaje indicando que falta Node.js.



Para solucionarlo:

1. Pulsa el botón: “Download Node.js”
2. Sublime Text descargará Node.js en la siguiente ubicación (el proceso puede tardar unos minutos, revisa la barra inferior):

```
~/config/sublime-text/Cache/LSP/Pyright/node-runtime
```

3. Reinicia Sublime Text. A partir de ese momento, al abrir cualquier módulo .py, el mensaje no volverá a aparecer.

7. Instalar SublimeLinter + linters

SublimeLinter es un plugin que añade un linter (herramienta) que aporta otra capa de análisis enfocada en **estilo de código y buenas prácticas**, funcionando como un “corrector ortográfico del código”.

Esto permite:

- Detectar errores de sintaxis sin ejecutar el programa.
- Avisar sobre malas prácticas y estilo según PEP8.
- Personalizar qué avisos mostrar o ignorar.

PEP8 es la **guía de estilo oficial de Python**, que establece un conjunto de normas y recomendaciones para escribir código de manera clara, legible y consistente.

Instalación de SublimeLinter

1. Ctrl+Shift+P → Install Package → SublimeLinter
2. Ctrl+Shift+P → Install Package → SublimeLinter-flake8
3. Instala flake8 **en el entorno virtual**, desde una terminal ejecutamos:

```
source ~/python-env/bin/activate
```

Con ello estamos cambiando la terminal para que python y pip apunten al entorno virtual, y que lo que instalemos quede en él y no afecte al sistema.

4. Instalamos flake8

Flake8 es un linter concreto para Python que combina comprobaciones de sintaxis y estilo según PEP8. SublimeLinter utiliza Flake8 para proporcionarte advertencias y errores en tiempo real.

```
pip install flake8
```

5. Sal del entorno virtual

```
deactivate
```

6. Abre Preferences → Package Settings → SublimeLinter → Settings.
7. Sustituye el texto del archivo por:

```
{
  "paths": {
    "linux": [
      "/home/user/python-env/bin"
    ]
  },
  "linters": {
  }
}
```

Donde user es el nombre de tu usuario.

8. Reinicia SublimeText

Personalizar los avisos de PEP8 de Flake8

Algunos avisos de Flake8 según PEP8 pueden resultar molestos, por ejemplo, las notificaciones de líneas demasiado largas (E501).

Para evitarlos, podemos configurar SublimeLinter para que ignore determinados códigos de error:

1. En Sublime Text, abre:
2. Preferences → Package Settings → SublimeLinter → Settings – User
3. Se abrirá el archivo que hemos editado en el punto anterior:
4. Dentro de la sección linters añade el código en rojo

```
{
  "paths": {
    "linux": [
      "/home/joses/python-env/bin"
    ]
  },
  "linters": {
    ""flake8": {
      "args": ["--ignore=E501"]
    }
  }
}
```

5. En este caso habrás introducido la opción de que no se muestren los avisos sobre líneas demasiado largas.
6. Puedes añadir otros códigos separados por comas. Por ejemplo, para ignorar también los espacios en blanco al final de una línea (W291):

```
"args": ["--ignore=E501,W291"]
```

8. Formatear código automáticamente con autopep8

autopep8 es una herramienta que analiza tu código Python y lo reformatea automáticamente para cumplir con las reglas de PEP8, la guía de estilo oficial de Python.

Para qué sirve

- Garantiza que tu código sigue las buenas prácticas de Python
- Corrige errores de formato de manera automática
- Facilita la lectura y el mantenimiento del código

Diferencias:

- **SublimeLinter te avisa:** mientras escribes, resalta los errores de sintaxis, los problemas de estilo y las malas prácticas según PEP8, pero no modifica el código. Te da la información para que decidas cómo corregirlo.
- **autopep8 te corrige:** toma tu código y aplica automáticamente los cambios necesarios para que cumpla con PEP8, ajustando espacios, indentación, saltos de línea y longitud de líneas. Esto te ahorra tiempo y asegura consistencia en el estilo.

Instalación

1. Si todavía no estás en el entorno virtual, abre una terminal y ejecuta:

```
source ~/python-env/bin/activate
```

2. Instala autopep8

```
pip install autopep8
```

3. Sal del entorno virtual

```
deactivate
```

Esto asegura que la instalación solo afecte a tu entorno de trabajo y no al sistema global.

Instalar y configurar el plugin en Sublime Text

Aunque autopep8 esté instalado en el entorno virtual, Sublime Text no lo detecta automáticamente. El plugin Python Autopep8 conecta Sublime con la herramienta instalada:

1. Abrir Command Palette (Ctrl+Shift+P)
2. Escribir Install Package → Enter
3. Buscar Python Autopep8 o AutoPEP8 → Enter

Ahora debes configurar la ruta al ejecutable dentro del entorno virtual:

4. Abre: Preferences → Package Settings → Python Autopep8 → Settings – User
5. Añade:


```
{
  "autopep8_path": "/home/joses/python-env/bin/autopep8",
  "args": ["--max-line-length", "79"]
}
```

6. Guarda los cambios.

Usar autopep8 en Sublime Text

1. Abre un archivo .py
2. Pulsa Ctrl+Shift+P → AutoPEP8: Format File
3. El código se reformatea automáticamente según PEP8

9. Configuración opcional del editor

Para ajustar Sublime Text a un estilo de edición más cómodo y consistente:

1. Ve a Preferences → Settings.
2. En la ventana de la derecha, sustituye el contenido por lo siguiente:

```
{
  "tab_size": 4,
  "translate_tabs_to_spaces": true,
  "detect_indentation": false,

  "highlight_line": true,
  "auto_indent": true,

  "save_on_focus_lost": true,
  "trim_trailing_white_space_on_save": true,

  "font_size": 16,
  "word_wrap": "auto"
}
```

Explicación de las opciones más importantes:

- tab_size: 4 espacios por tabulación.
- translate_tabs_to_spaces: convierte tabs en espacios automáticamente.
- detect_indentation: desactiva la detección automática de indentación.
- highlight_line: resalta la línea actual donde está el cursor.
- auto_indent: aplica indentación automática al crear nuevas líneas.
- save_on_focus_lost: guarda el archivo automáticamente al cambiar de ventana.
- trim_trailing_white_space_on_save: elimina espacios en blanco al final de las líneas.
- font_size: tamaño de letra del editor.
- word_wrap: ajusta el texto automáticamente al ancho de la ventana.

10. Instalar SideBarEnhancements

SideBarEnhancements es un complemento que mejora las opciones del Sidebar de Sublime Text, facilitando la gestión de archivos y carpetas.

Instalación:

1. Ve a Preferences → Package Control.
2. Escribe Install Package y pulsa Enter.
3. Busca SideBarEnhancements y selecciona para instalar.

Uso:

Tras la instalación, haz clic derecho sobre cualquier archivo o carpeta en el Sidebar.

Verás todas las opciones adicionales que ofrece el complemento para copiar, mover, renombrar, abrir terminal, y más.

11. Añadir iconos según tipo de archivo

Para mejorar la apariencia de los iconos de los archivos en el Sidebar:

Instalación:

1. Ve a Preferences → Package Control (Ctrl+Shift+P).
2. Selecciona Install Package y pulsa Enter.
3. Busca A File Icon y haz clic para instalar.

Resultado:

Los archivos mostrarán iconos más atractivos y representativos según su tipo.

12. Anexo – Desinstalar Sublime Text en Linux Mint

Antes de realizar la instalación de la versión descrita en este tutorial sería recomendable desinstalar cualquier resto que hubiera de instalaciones anteriores. Para ello seguiremos los siguientes pasos:

Desinstalar Sublime Text

Abre la terminal y ejecuta:

```
sudo apt remove --purge sublime-text
```

- remove elimina el paquete principal.
- --purge también elimina los archivos de configuración instalados por el paquete (los que están en /etc/apt/ relacionados con Sublime).

Luego puedes limpiar paquetes residuales:

```
sudo apt autoremove
```

Eliminar configuraciones de usuario y caché

Sublime guarda configuraciones y paquetes instalados en tu directorio de usuario, no se eliminan con apt remove. Para borrarlas:

```
rm -rf ~/.config/sublime-text-3  
rm -rf ~/.config/sublime-text  
rm -rf ~/.cache/sublime-text-3  
rm -rf ~/.cache/sublime-text
```

En Linux Mint, los nombres de carpeta pueden variar según la versión:

Sublime Text 3 → sublime-text-3

Sublime Text 4 → sublime-text

 Esto elimina todos tus paquetes, settings y proyectos.

Verificar que todo se borró

```
which subl
```

Si no devuelve nada, Sublime Text ya no está en el sistema.

También puedes buscar configuraciones residuales:

```
ls ~/.config | grep sublime  
ls ~/.cache | grep sublime
```