

# Actividad Extracurricular04B

January 19, 2026

## 1 Escuela Politécnica Nacional

### 1.1 Métodos Numéricos

Nombre: Lenin Amangandi

### 1.2 Probar diferentes formatters populares en python.

#### 1) Black Formatter

Black es un formateador de código para Python diseñado para hacer que el código sea fácil de leer sin que los desarrolladores tengan que preocuparse por los detalles de estilo. Se basa en la filosofía de que “hay una forma correcta de escribir el código”, lo que significa que no hay muchas opciones configurables, lo que lo hace rápido y sencillo de usar.

```
# Antes de Black
def saludo( nombre ):
    print("Hola " + nombre)
```

```
# Después de Black
def saludo(nombre):
    print("Hola " + nombre)
```

#### 2) autopep8

autopep8 también sigue las reglas de PEP 8, pero posee más flexibilidad. A diferencia de Black, se configura para que solo corrija ciertos errores de estilo y mantenga otras cosas como están. Es ideal si para mantener el código limpio sin que se cambie todo el estilo automáticamente. Se puede personalizar y elegir qué tipos de correcciones hacer, lo que te da un control extra sobre el proceso.

```
# Antes de autopep8
def saludo( nombre ):
    if nombre == "Lenin":
        print( ";Hola Lenin!" )
    else:
        print(";Hola!")
# Después de Black
def saludo(nombre):
    if nombre == "Lenin":
        print(";Hola Lenin!")
```

```
else:
    print("¡Hola!")
```

### 1.3 3) YAPF

YAPF (Yet Another Python Formatter) es un formateador que sigue las reglas de PEP 8 pero con un enfoque más flexible que autopep8. Permite personalización y ajusta el código a un estilo consistente, pero con más control sobre cómo y qué cambios realizar. Es útil cuando deseas aplicar un formato que no sea tan estricto como el de Black pero sin perder la coherencia del código. Puedes configurarlo para que se adapte mejor a tus preferencias o las de tu equipo.

```
“python # Antes de YAPF def saludo(nombre): if nombre=="Lenin": print("¡Hola, Lenin!");else:
print("¡Hola!") # Después de YAPF def saludo(nombre): if nombre == "Lenin": print("¡Hola,
Lenin!") else: print("¡Hola!")
```

### 1.4 3) isort

**isort** es una herramienta para ordenar las importaciones en un archivo Python, asegurando que sigan un estilo coherente y estén organizadas de acuerdo con las convenciones de PEP 8. A diferencia de otros formateadores como Black o autopep8, que se enfocan en el formato general del código, **isort** se especializa únicamente en las importaciones, ayudando a que tu código sea más limpio y fácil de leer. Es muy útil cuando tienes muchas importaciones en tus archivos y necesitas organizarlas sin tener que hacerlo manualmente.

#### Características de isort:

- **Ordena las importaciones:** Organiza automáticamente las importaciones en grupos según el tipo de importación (estándar, de terceros, locales, etc.).
- **Cumple con PEP 8:** Sigue las reglas de PEP 8 para las importaciones, lo que significa que ordena las importaciones alfabéticamente y las separa en bloques de manera adecuada.
- **Configuración flexible:** Puedes configurarlo para que se adapte a tus preferencias específicas, como el tamaño de la separación entre los bloques de importación o si deseas que ciertas importaciones estén en un orden específico.
- **Soporte para proyectos grandes:** isort es especialmente útil cuando trabajas en proyectos grandes donde las importaciones se acumulan y se vuelven difíciles de mantener.

#### Ejemplo: Antes de isort:

```
import os
import sys
import math
```

#### Despues de isort:

```
import math
import os
import sys
```

#### Ejemplo de Configuración:

```
[settings]
profile = black
```

```
line_length = 88
known_third_party = numpy, pandas
```

Esto configura isort para seguir el mismo estilo de black, establece una longitud de línea de 88 caracteres, y agrupa las importaciones de numpy y pandas en la categoría de importaciones de terceros.

#### 1.4.1 Cuándo utilizarlo:

- **Proyectos grandes:** Cuando se tiene muchas importaciones en los archivos Python, isort es ideal para mantenerlas organizadas y ordenadas sin esfuerzo manual.
- **Colaboración en equipo:** Ayuda a mantener un estilo coherente en las importaciones en todo el proyecto, lo que es muy útil cuando se trabaja con otras personas.
- **Complemento a otros formateadores:** Se puede usar junto con Black o autopep8, ya que isort se enfoca únicamente en las importaciones y no cambia otros aspectos del código.