



**Metodología educación a distancia**

**Programación avanzada**

**Corte 1**

**Jefferson David Rios Moreno**

**Codigo:2286816**

**Docente**

**David Bohórquez**

**Universidad Santo Tomas**

**Decantara de división de educación abierta y a distancia**

**Ingeniería informática**

**Centro de atención universitario Duitama**

**Duitama**

**2023-2**



## Tabla de contenido

1. Introducción.....	3
2. Objetivos.....	4
2.1. Objetivos generales: .....	4
2.2. Objetivos específicos: .....	4
3. Desarrollo Actividad 1.....	5
4. Conclusiones.....	7



## 1. Introducción

PEP 8 es un documento que establece las convenciones de estilo para el código Python. Su objetivo es mejorar la legibilidad y consistencia del código Python.



## 2. Objetivos

### 2.1. Objetivos generales:

- Mejorar la legibilidad del código Python.
- Hacer que el código Python sea más fácil de entender y mantener.
- Promover la consistencia del código Python.

### 2.2. Objetivos específicos:

- Establecer reglas para la longitud de las líneas, el espaciado y la indentación.
- Establecer reglas para el nombre de las variables, funciones y clases.
- Establecer reglas para el uso de comentarios y documentación.



### 3. Desarrollo Actividad 1

PEP 8 es una guía de estilo para el código de Python. Su objetivo es mejorar la legibilidad, la consistencia y la mantenibilidad del código.

#### - Identificación

PEP 8 recomienda usar cuatro espacios para la sangría. Los espacios en blanco deben ser uniformes en todo el código.

#### Ejemplo

```
#Esto crea una funcion que toma dos argumentos
def my_funcion(a, b):
    return a + b
```

#### - Espacios en blanco

PEP 8 recomienda usar espacios en blanco alrededor de los operadores aritméticos, asignación y comparación.

#### Ejemplo

```
#Contiene los espacios correctos para las operaciones
a = 1 + 2
b = a * 3
c = a == b
```

#### - Líneas y espacios

PEP 8 recomienda limitar el ancho de las líneas a 79 caracteres. Las líneas más largas deben ser divididas en varias líneas usando un carácter de continuación.



### Ejemplo:

```
#funcion con muchos argumentos todos separados correctamente
def my_funcion(a, b, c, d, e, f, g, h, i, j, k, l, o):
    return a + b + c + d + e + f + g + h + i + j + k + l + o
```

#### - Nombres

PEP 8 recomienda usar nombres de variables, funciones y clases que sean descriptivos y fáciles de entender.

### Ejemplo:

```
#Crea una funcion que suma 2 numeros
def suma (a, b):
    return a + b
#Crea una clase que representa una persona
class Persona(object):
    def __init__(self, nombre, apellido, edad):
        self.nombre = nombre
        self.apellido = apellido
        self.edad = edad
```

#### - Importaciones

Por lo general, las importaciones deben realizarse en líneas separadas

### Ejemplo

```
import pandas as pd
import os
import sys
from subprocess import Popen, PIPE
```



#### 4. Conclusiones

PEP 8 es una guía importante para los desarrolladores de Python. Al seguir estas convenciones, los desarrolladores pueden escribir código que sea más fácil de leer, entender y mantener.

Algunos ejemplos de las reglas de PEP 8:

- Las líneas de código deben tener un máximo de 79 caracteres.
- Se debe utilizar un espacio para indentar el código.
- Los nombres de las variables deben ser significativos y descriptivos.
- Los comentarios deben ser completos y precisos.



## 5. Bibliografías

<https://peps.python.org/pep-0008/>