

**Centro Educativo Técnico Laboral Kinal**  
**Área de Diversificado**  
**Tecnología**



***Estilos de codificación***

**Presentado Por:**  
**Ander Ezequiel Cabrera Calel**  
**Grado «Sexto Perito»**  
**Sección «A»**  
**Jornada «V»**

**Guatemala, mayo de 2024**

## **Introducción:**

En el desarrollo de software, no se limita simplemente a escribir código. Más bien, es una ciencia que requiere precisión, claridad y coherencia. Aquí es donde entran en juego los estilos de codificación, también conocidos como guías de estilo de código. Estos documentos, que a menudo son pasados por alto, pero de gran importancia, establecen las convenciones y normas para escribir código en un lenguaje de programación específico.

Los estilos de codificación no son simplemente un conjunto de reglas aleatorias; más bien, son reglas diseñadas para mejorar la legibilidad, mantenibilidad y colaboración dentro de un proyecto de software. Desde la indentación hasta la capitalización de variables, desde la longitud de las líneas hasta la documentación del código, cada aspecto del estilo de codificación está cuidadosamente considerado para garantizar que el código sea fácilmente comprensible y modificable por otros desarrolladores.

A lo largo de los años, diferentes comunidades de desarrollo han desarrollado sus propias guías de estilo, reflejando las preferencias y mejores prácticas de cada grupo. Estos estilos pueden variar significativamente entre sí, pero comparten el objetivo común de promover la coherencia y la calidad del código. Desde el "PEP 8" de Python hasta el "Google Java Style Guide" y el "Airbnb JavaScript Style Guide", cada uno ofrece una hoja de ruta clara para escribir código limpio y eficiente.

En esta investigación, veremos la importancia de estas guías, los principios de cada una y cómo pueden impactar positivamente en la productividad y la calidad del código. Además, veremos algunos ejemplos de diferentes guías de estilo en algunos lenguajes de programación.

# Indice

<b>¿Que son los estilos de codificación?</b>	<b>4</b>
<b>Ventajas</b>	<b>4</b>
<b>Python</b>	<b>5</b>
<b>Javascript</b>	<b>7</b>
<b>Conclusión</b>	<b>9</b>
<b>Referencias</b>	<b>10</b>

## ¿Que son los estilos de codificación?

Los estilos de codificación, también conocidos como guías de estilo de código, son conjuntos de convenciones y reglas que establecen cómo se debe escribir el código en un lenguaje de programación específico. Estas guías definen aspectos como la indentación, la capitalización de variables, la longitud de las líneas, la documentación del código y otras convenciones de formato.

## Ventajas

- **Mejora de la legibilidad:** Los estilos de codificación establecen reglas claras sobre el formato y la estructura del código, lo que facilita su comprensión para los desarrolladores. Una sintaxis coherente y bien organizada hace que el código sea más fácil de entender, incluso para aquellos que no lo escribieron originalmente.
- **Facilita la colaboración:** Al seguir un estilo de codificación consistente, los equipos de desarrollo pueden colaborar de manera más efectiva. Todos los miembros del equipo saben qué esperar en términos de estructura, lo que facilita la revisión, modificación y fusión del código de manera conjunta sin conflictos innecesarios.
- **Reducción de errores y mejor mantenibilidad:** Los estilos de codificación ayudan a prevenir errores comunes y fallos de programación al establecer reglas claras sobre cómo escribir código. Además, la consistencia en el estilo facilita la identificación y corrección de errores durante el mantenimiento del código, lo que contribuye a una arquitectura más robusta y a un software más confiable a largo plazo.

## Python

PEP 8 es una guía de estilo de Python que establece reglas para la escritura de código en Python con el objetivo de mejorar la legibilidad y la consistencia del mismo.

Ejemplos de algunas reglas de PEP 8:

**Convenciones de nombres:** Hay que utilizar nombres descriptivos para variables, funciones y clases. Para variables, hay que utilizar minúsculas con palabras separadas por guiones bajos (snake case).

```
main.py
1  # Mal
2  myvariable = 5
3
4  # Bien
5  my_variable = 5
```

**Indentación:** Hay que utilizar cuatro espacios para la indentación en lugar de tabulaciones.

```
main.py
1  # Mal
2  def my_function():
3      \tprint("Hello")
4
5  # Bien
6  def my_function():
7      print("Hello")
8
```

**Longitud máxima de línea:** Limita la longitud de las líneas de código a 79 caracteres.

```
main.py
1  # Mal
2  my_long_variable_name = "This is a very long string that exceeds the
    maximum line length recommended by PEP 8"
3
4  # Bien
5  my_long_variable_name = "This is a string that fits within the
    recommended maximum line length of PEP 8"
6
7
```

**Espacios en blanco:** Utiliza espacios alrededor de operadores y después de comas, pero no en paréntesis de expresiones.

```
main.py
1  # Mal
2  result=a+b
3
4  # Bien
5  result = a + b
6
7  |
```

## Javascript

La guía de estilo de Airbnb para JavaScript establece una serie de convenciones para escribir código en JavaScript que enfatiza la legibilidad, la claridad y la consistencia.

Aquí hay algunos ejemplos de la guía de estilos de Airbnb:

**Convenciones de nombres:** Utiliza camelCase para los nombres de variables y funciones.

```
main.js
1  // Mal
2  let my_variable = 5;
3
4  // Bien
5  let myVariable = 5;
6
```

**Indentación:** Utilizar dos espacios para la indentación.

```
main.js
1  // Mal
2  function myFunction() {
3      console.log("Hello");
4  }
5
6  // Bien
7  function myFunction() {
8      console.log("Hello");
9  }
10
```

**Uso de punto y coma:** Utilizar punto y coma al final de cada declaración.

```
main.js
1 // Mal
2 const x = 5
3 const y = 10
4
5 // Bien
6 const x = 5;
7 const y = 10;
8
```

**Declaración de variables:** Utilizar let o const en lugar de var para declarar variables.

```
main.js
1 // Mal
2 var myVariable = 5;
3
4 // Bien
5 const myVariable = 5;
6
```

**Declaración de funciones flecha:** Utilizar funciones flecha en lugar de funciones anónimas cuando sea posible.

```
main.js
1 // Mal
2 function sum(a, b) {
3   return a + b;
4 }
5
6 // Bien
7 const sum = (a, b) => {
8   return a + b;
9 };
10
```



## **Conclusión**

En conclusión, tanto en Python con la guía de estilo PEP 8 como en JavaScript con la guía de estilo de Airbnb, adoptar y seguir las convenciones y reglas establecidas puede tener un impacto significativo en la calidad, la legibilidad y la mantenibilidad del código.

Las guías de estilo no son simplemente sugerencias arbitrarias, sino que promueven la consistencia y la claridad en el código. Al seguir estas reglas, los desarrolladores pueden crear un código más limpio, fácil de entender y menos propenso a errores. Además, las guías de estilo facilitan la colaboración entre diferentes miembros del equipo, ya que todos están escribiendo código de la misma manera, lo que reduce la confusión en el proceso de desarrollo de software.

También se toma en cuenta que cada empresa puede tener sus propias guías de estilo para los diferentes lenguajes de programación que utilicen y es importante que siempre nos adaptemos a ellas.

## Referencias

Airbnb. (2024). *Github*. Obtenido de Javascript: <https://github.com/airbnb/javascript>

Rossum, G. v., Barry Warsaw, & Alyssa Coghlan . (05 de 07 de 2001). *python*. Obtenido de PEP 8: <https://peps.python.org/pep-0008/>