

**Definir Estándares de Codificación de Acuerdo a Plataforma de Desarrollo Elegida**

Presentado por Karen Julieth Sosa Espitia

Análisis y desarrollo de software

2977481

Presentado a Instructor Milton Ivan Barbosa Gaona

Centro de la Tecnología del Diseño y la Productividad Empresarial Girardot –

Cundinamarca – SENA (Servicio Nacional de Aprendizaje)

28 de Julio 2025

## Tabla de contenido

Introducción.....	4
Justificación.....	5
Objetivos.....	6
Objetivo General.....	6
Objetivos Específicos.....	6
1. Estandares de Codificación en Java.....	7
1.1 Convenciones de Nombres.....	7
1.1.1 Clases.....	7
1.1.2 Métodos.....	7
1.1.3 Variables.....	7
1.1.4 Constantes.....	7
1.2 Formato del código.....	7
1.3 Declaración de una variable.....	8
1.4 Declaración de Métodos.....	8
1.5 Declaración de clases.....	8
1.6 Buenas prácticas.....	8
Conclusiones.....	9
Bibliografía.....	10

### **Contenido de Imágenes**

No se encuentran elementos de tabla de ilustraciones.

## **Introducción**

En este documento se presenta los estándares de codificación del lenguaje de programación Java, desde las convenciones (conjunto de reglas y buenas prácticas) de los nombres, las estructuras de las declaraciones, formato del código y buenas prácticas en general, con la finalidad de tener un código claro, consistente y fácil de mantener.

### **Justificación**

La programación en Java exige precisión, organización y claridad, lo cual solo se logra mediante una correcta declaración de elementos y el seguimiento de convenciones estandarizadas, es crucial que se comprenda cómo aplicar buenas prácticas de codificación. Este conocimiento no solo mejora el rendimiento individual, sino que también facilita el trabajo colaborativo y la escalabilidad de los proyectos.

## **Objetivos**

### **Objetivo General**

Consultar los estandares de codificación de Java

### **Objetivos Específicos**

- Especificar las convenciones de nombres
- Entender la declaración de variables, metodos y clases en Java
- Describir las buenas prácticas

## **1. Estándares de Codificación en Java**

### **1.1 Convenciones de Nombres**

#### **1.1.1 Clases**

Para las clases , la primera letra siempre debe ser en mayúscula, uso de nomenclatura camelCase, y deben ser sustantivos. De la siguiente manera: NombreDeLaClase.

#### **1.1.2 Métodos**

Se hace uso de camelCase, es decir, la primera letra de la primera palabra en minúscula, y el resto de palabras inicia con mayúscula, tienen que ser verbos, y se puede agregar sustantivos. Un método solo debe hacer una cosa. Ejemplo: revisarCuenta.

#### **1.1.3 Variables**

Se hace uso del camelCase, no se debe iniciar con número ni caracteres especiales, no puede tener el mismo nombre que una palabra clave, y no debe tener el mismo nombre de otras variables cuyas declaraciones aparezcan en el mismo ámbito. Ejemplo: numeroFactura

#### **1.1.4 Constantes**

Tiene que escribirse en mayúsculas, y separadas las palabras por guion bajo. Como por ejemplo: MAX\_INTENTOS.

### **1.2 Formato del código**

Usar sangría de 4 espacios por nivel (no tabulaciones), limitar la longitud de línea a 80-120 caracteres para mejor lectura, y usar espacios después de comas y operadores. (OpenAI, 2025)

### 1.3 Declaración de una variable

Se define que tipo de dato va a almacenar ( float, double,char,string, boolean,etc) y el nombre que tendrá la variable.También se puede declarar y asignar en la misma fila. Ejemplo:

- int edad
- String apellido

### 1.4 Declaración de Métodos

Define en el siguiente orden: el modificador de acceso, que tipo de valor devuelve, que nombre tendrá y que parámetros necesita (puede no tener).

Por ejemplo:

```
public int obtenerEdad () {
    return edad;
}
```

### 1.5 Declaración de clases

```
public class NombreDeLaClase {}
```

Donde Public es el modificador de acceso, la palabra class indica que es una Clase, sigue el nombre de la clase y las llaves se usan para agrupar bloques de código, donde van sus atributos y metodos.

### 1.6 Buenas prácticas

- Comentar cada 3-7 líneas del código, solo lo necesario
- Utilice nombres descriptivos y adecuados para todos los identificadores (variables, nombres de métodos, nombres de clases constantes, etc. (*University of Hawai'i at Mānoa, s. f.*)
- No utilice guiones ni guiones bajos para separar identificadores de varias palabras (excepto las constantes) (*University of Hawai'i at Mānoa, s. f.*)
- Dividir el código en métodos pequeños y reutilizables y evitar duplicación de código.



## **Conclusiones**

- El uso de estándares de codificación mejora la calidad del código, haciéndolo más legible , menos propenso a errores, además de ayudar a favorecer la comprensión en un trabajo en equipo.
- La declaración en Java es esencial puesto que es la base para que el código sea comprensible y funcional.
- Los métodos deben declararse con una estructura clara que permita comprender rápidamente qué hace el método y cómo usarlo.

### **Bibliografia**

OpenAI. (2025). *ChatGPT (versión GPT-4o)* [Modelo de lenguaje grande].

<https://chat.openai.com/>

University of Hawai'i at Mānoa. (s. f.). *Java coding standards*. En *ICS 211: Materials*.

Recuperado el 28 de julio de 2025, de

<https://www2-hawaii-edu.translate.goog/~walbritt/ics211/materials/standard.htm>