

Facultad de matemáticas

Estándar de conteo de líneas de código físicas y lógicas

Versión 1.1

Autores:

Aaron Isaac Graniel Arzat

Fernando Joachín Prieto

David Peña Muñoz

Jose Luis Pooc Moo

Andrea Isabel Torres Perez

**CONTROL DE DOCUMENTACIÓN**

**Control de Configuración**

| **Título:** | Estándar de conteo de líneas de código físicas y lógicas |
| --- | --- |
| **Referencia:** | ESP\_EstandarConteoLineasCodigo\_v1.0\_2025-02-17.docx |
| **Autor(es):** | Joachin Fernando Joachin |
| **Fecha:** | 17/02/2025 |

**Histórico de versiones**

| **Versión** | **Fecha** | **Estado** | **Responsable** | **Nombre de archivo** |
| --- | --- | --- | --- | --- |
| 1.0 | 17/02/2025 | A | Joachin Prieto Fernando | ESP\_EstandarConteoLineasCodigo\_v1.0\_2025-02-17 |

**Estado:** (B)orrador, (R)evisión, (A)probado

**Histórico de cambios**

| **Versión** | **Fecha** | **Cambios** |
| --- | --- | --- |
| 1.0 | 17/02/2025 | Ninguna, primera versión borrador. |
| 1.0 | 28/02/2025 | Actualización del estándar de conteo de líneas lógicas en Java, ajuste alineado con la clarificación del profesor para priorizar elementos estructurales del código. |

**Índice**

[**1. Introducción 4**](#_ggu3rbepykw9)

[**2. Alcance 4**](#_o63kukkgzih4)

[**3. Propósito 4**](#_h6v7kthy37n4)

[**4. Aplicación 5**](#_8op0yyf6l9wa)

[**5. Definición de Líneas 5**](#_h9086b9zwjt6)

[5.1 Líneas lógicas 5](#_7u9abnxs3lnt)

[5.2 Líneas físicas 5](#_963zizmf0p6d)

[**6. Reglas de conteo 6**](#_ww2gx6xgevll)

[6.1 Líneas lógicas 6](#_ttdwi2gc88z9)

[6.1.1 Líneas que se incluyen en el conteo 6](#_n4ulffgy904g)

[6.1.1.1 Declaraciones de estructuras fundamentales 6](#_6rftn19p72u0)

[6.1.1.2 Declaraciones estructurales de control 6](#_5x43bdex7f2q)

[6.1.1.3 Anidamiento de estructuras de control 6](#_jc4jfgdxwyhe)

[6.1.2 Líneas que no se incluyen en el conteo 7](#_8yrv44lfix5q)

[6.1.2.1 Sentencias ejecutables 7](#_jkrpzkjvs93f)

[6.1.2.2 Comentarios únicos 7](#_xnwr0a7p7qrp)

[6.1.2.3 Espacios en blanco 7](#_3pxkk81cmagz)

[6.1.2.4 Delimitadores sin código 7](#_csabaz60owr)

[6.1.2.5 Anotaciones 7](#_jcmksp70vbki)

[6.1.2.6 Importaciones 7](#_lp2lx48p6g7o)

[6.1.2.7 Elementos internos de estructuras de control 7](#_p59kipb30f0r)

[6.2 Líneas físicas 8](#_hgbtmuu4xalb)

[6.2.1 Líneas que se incluyen en el conteo 8](#_80uxlb5bc8mq)

[6.2.1.1 Fragmentos de código 8](#_wc6qx0fi2hx)

[6.2.1.2 Presencia en el archivo 8](#_pbkd8v2he97o)

[6.2.2 Líneas que no se incluyen en el conteo 8](#_xxlqt54zxg8y)

[6.2.2.1 Líneas en blanco 8](#_iqpztsemjb4w)

[6.2.2.2 Líneas solo con comentarios 8](#_l45hsxs7is9h)

[**7. Anexo A. Guía ilustrativa de las reglas de conteo 9**](#_td19mk9as8p)

# 

# Introducción

Este documento define un estándar de conteo de líneas de código (LOC) para programas en Java, estableciendo criterios claros sobre qué se considera y qué se excluye en el conteo.

# Alcance

El alcance de este documento abarca la definición y aplicación de un estándar de conteo de líneas de código (LOC) para programas desarrollados en el lenguaje de programación Java. El estándar se enfoca en establecer criterios claros y consistentes para contar tanto las líneas de código físicas como las lógicas, con el fin de garantizar la uniformidad en la medición en proyectos de software.

Este estándar es aplicable a todos los proyectos de desarrollo de software en Java dentro de la organización, independientemente de su tamaño o complejidad. Incluye la definición de qué tipos de líneas deben ser contadas y cuáles deben ser excluidas.

# Propósito

El propósito principal de este estándar es definir una metodología clara y consistente para realizar el conteo de líneas de código físicas y lógicas en programas desarrollados en Java. Este sistema de conteo tiene como objetivo:

1. **Uniformidad**: Establecer criterios unificados para contar líneas de código, asegurando que todos los equipos y proyectos sigan las mismas reglas y eviten discrepancias en las mediciones.
2. **Precisión**: Proporcionar directrices específicas sobre qué se considera una línea física y una línea lógica, así como qué elementos deben incluirse o excluirse en el conteo (por ejemplo, comentarios, líneas en blanco, etc.).
3. **Transparencia**: Facilitar la comprensión y replicabilidad del proceso de conteo, permitiendo que cualquier miembro del equipo o auditor pueda realizar la medición de manera consistente.
4. **Eficiencia**: Simplificar el proceso de conteo de líneas de código, reduciendo la subjetividad y el tiempo invertido en debates sobre qué debe o no contarse.

En resumen, este estándar busca ser una herramienta práctica y objetiva para realizar el conteo de líneas de código físicas y lógicas, sin involucrar aspectos de complejidad o calidad del código, sino centrándose únicamente en la medición precisa y consistente del volumen de código.

# Aplicación

Este estándar de conteo de líneas de código físicas y lógicas es aplicable a cualquier proyecto desarrollado en Java que siga el estándar de codificación establecido por la organización (ESP\_EstandarCodificacionProyectoConteoLineas\_v1.0\_2025-02-17). Su uso garantiza que el conteo de líneas de código se realice de manera consistente y alineada con las prácticas de desarrollo definidas, independientemente de la fase del proyecto (desarrollo nuevo, mantenimiento o modificación de código existente).

# Definición de Líneas

Un programa en Java se distingue entre un número de líneas lógicas y un número de líneas físicas.

## 5.1 Líneas lógicas

Una línea lógica en Java se refiere a una línea de código que establece la estructura fundamental del programa, es decir, aquellas que definen su arquitectura, organización o flujo de control. Esto incluye las declaraciones de clases, interfaces, enumeraciones, métodos y constructores, así como las estructuras de control que afectan el flujo del programa, tales como if, for, while, do-while, switch y try-catch. En el caso de las estructuras de control, solo se cuenta la línea donde se declara la estructura, sin considerar su contenido. Si una estructura de control contiene otra dentro de ella, cada nueva estructura se cuenta como una línea lógica adicional.

## 5.2 Líneas físicas

Una línea física en Java es una secuencia de caracteres terminada por una secuencia de final de línea. En los archivos fuente, se pueden utilizar cualquiera de las convenciones estándar de terminación de línea de la plataforma: el formato Unix con ASCII LF (\n), el formato Windows con ASCII CR LF (\r\n), o el formato antiguo de Macintosh con ASCII CR (\r). Java es independiente de la plataforma en cuanto a la interpretación de líneas en el código fuente.

Cuando se trabaja con entradas de texto o flujos de datos, la clase BufferedReader y otros mecanismos de lectura en Java manejan las diferencias entre terminaciones de línea de forma transparente. Además, en las cadenas de texto dentro del código fuente, el carácter \n representa un salto de línea en conformidad con la convención estándar de Java y Unicode.

# 

# Reglas de conteo

## 6.1 Líneas lógicas

### 6.1.1 Líneas que se incluyen en el conteo

#### 6.1.1.1 Declaraciones de estructuras fundamentales

Se incluyen las declaraciones de clases, interfaces, enumeraciones, métodos y constructores, ya que establecen la arquitectura y organización del programa.

#### 6.1.1.2 Declaraciones estructurales de control

Se consideran las estructuras de control que afectan el flujo del programa, tales como if, for, while, do-while, switch, try y try-catch, pero sólo la línea donde se declara la estructura de control cuenta como línea lógica. El contenido dentro de estas estructuras no se considera en el conteo.

#### 6.1.1.3 Anidamiento de estructuras de control

Si una estructura de control contiene otra estructura de control dentro de ella, cada nueva estructura cuenta como una línea lógica adicional. Por ejemplo, un if dentro de un for se contaría como dos líneas lógicas.

### 

### 6.1.2 Líneas que no se incluyen en el conteo

#### 6.1.2.1 Sentencias ejecutables

Las sentencias de asignación, invocación de métodos y operaciones aritméticas no se consideran líneas lógicas, ya que no definen la estructura fundamental del programa.

#### 6.1.2.2 Comentarios únicos

Se excluyen las líneas que contienen únicamente comentarios, ya sean comentarios de línea (iniciados por //), comentarios en bloque (/\* ... \*/) o comentarios Javadoc (/\*\* ... \*/).

#### 6.1.2.3 Espacios en blanco

No se cuentan las líneas que sólo contienen espacios, tabulaciones u otros caracteres de espaciado sin contenido ejecutable.

#### 6.1.2.4 Delimitadores sin código

Las líneas que contienen únicamente delimitadores de bloque (como { o }) sin estar asociadas directamente a una sentencia ejecutable no se consideran líneas lógicas, salvo cuando forman parte integral de una declaración estructural.

#### 6.1.2.5 Anotaciones

Las anotaciones en Java (como @Override, @Deprecated, @SuppressWarnings, etc.) son metadatos que proporcionan información adicional sobre el código, pero no son instrucciones ejecutables. Por lo tanto, no cuentan como líneas lógicas en el análisis.

#### 6.1.2.6 Importaciones

Las declaraciones de importación en Java (import java.util.List;, import static java.lang.Math.PI;, etc.) permiten el uso de clases y métodos de otros paquetes sin necesidad de escribir su nombre completo. Dado que las importaciones solo afectan la accesibilidad de las clases y no contienen lógica ejecutable, no se consideran líneas lógicas en el conteo del código.

#### 6.1.2.7 Elementos internos de estructuras de control

Las sentencias dentro de if, for, while, do-while, switch, try y try-catch no cuentan como líneas lógicas, con excepción de cuando una estructura de control contiene otra estructura de control dentro de ella. En el caso de switch, solo se cuenta la línea de declaración del switch, mientras que los case y default no se consideran líneas lógicas.

## 6.2 Líneas físicas

### 6.2.1 Líneas que se incluyen en el conteo

#### 6.2.1.1 Fragmentos de código

Se incluyen todas las líneas del archivo fuente que contengan cualquier parte de código ejecutable, incluso si la línea también incluye comentarios, siempre que exista código identificable.

#### 6.2.1.2 Presencia en el archivo

Cada secuencia de caracteres que termina con un delimitador de línea y que contiene fragmentos de instrucciones se cuenta como una línea física, sin importar su extensión o formato.

### 6.2.2 Líneas que no se incluyen en el conteo

#### 6.2.2.1 Líneas en blanco

Se excluyen las líneas que están completamente vacías o que contienen únicamente espacios o tabulaciones.

#### 6.2.2.2 Líneas solo con comentarios

No se consideran las líneas en las que el único contenido son comentarios, ya sean de línea (//) o bloques de comentarios (/\* ... \*/ o /\*\* ... \*/).

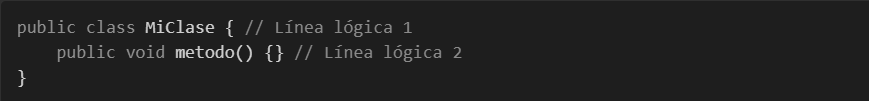
# 

# Anexo A. Guía ilustrativa de las reglas de conteo

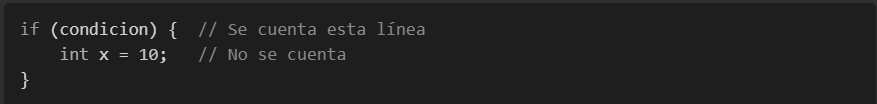
**7.1 Líneas lógicas**

**7.1.1 Líneas que se incluyen en el conteo**

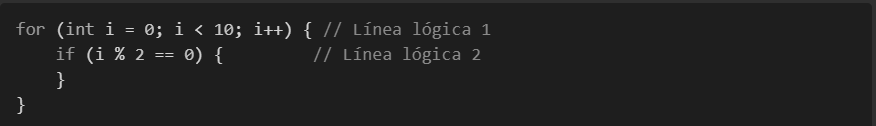
**7.1.1.1 Declaraciones de estructuras fundamentales**

****

**7.1.1.2 Declaraciones estructurales de control**

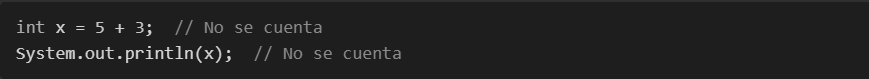
****

**7.1.1.3 Anidamiento de estructuras de control**

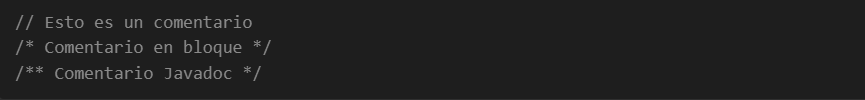
****

**7.1.2 Líneas que no se incluyen en el conteo**

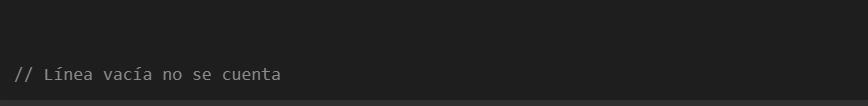
**7.1.2.1 Sentencias ejecutables**



**7.1.2.2 Comentarios únicos**



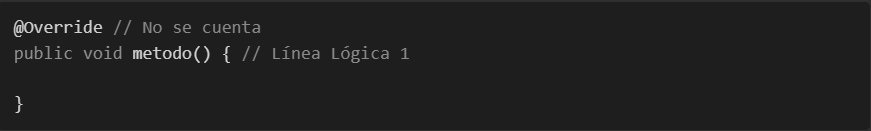
**7.1.2.3 Espacios en blanco**

****

**7.1.2.4 Delimitadores sin código**

****

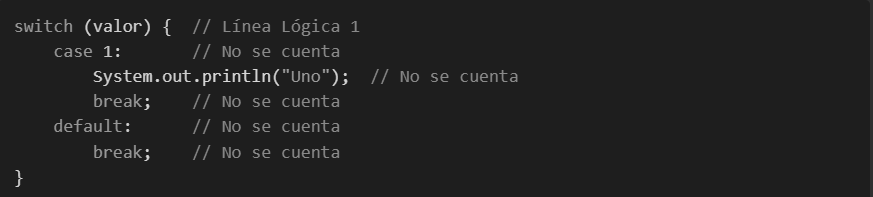
**7.1.2.5 Anotaciones**

****

**7.1.2.6 Importaciones**

****

**7.1.2.7 Elementos internos de estructuras de control**

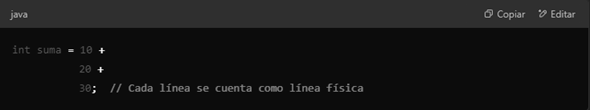
****

**6.2 Líneas físicas**

**6.2.1 Líneas que se incluyen en el conteo**

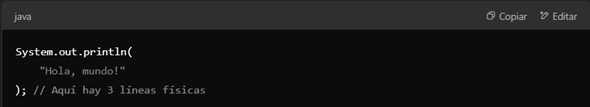
**6.2.1.1 Fragmentos de código**





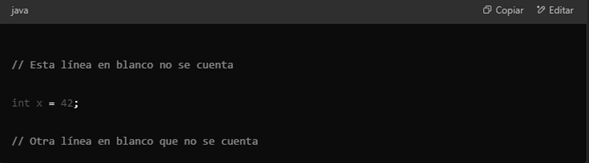
**6.2.1.2 Presencia en el archivo**





**6.2.2 Líneas que no se incluyen en el conteo**

**6.2.2.1 Líneas en blanco**



**6.2.2.2 Líneas solo con comentarios:**

