

MANUAL DE USUARIO

Equipo 2

14 de mayo de 2025

Realizó Mantenimiento:

Canul Ordoñez, Josué Israel

Garcilazo Cuevas, Mónica

Leo Fernández, José Carlos

Pool Flores, Endrick Alfredo

Rodríguez Coral, Samuel David

1 CONTROL DE DOCUMENTACIÓN

1.1 Control de configuración

Título:	Manual de usuario
Referencia:	ISO/IEC 26514
Autor:	Isaac Ehuan
Fecha:	26 de febrero de 2025

1.2 Histórico de versiones

Versión	Fecha	Estado	Responsable	Nombre del archivo
1.0.1	26 – 02 – 2025	B	Isaac Ehuan	Manual_de_usuario_1 .0.1
1.0.2	27 – 02 – 2025	A	Paulina Perera	Manual_de_usuario_1 .0.2
2.0.0	02 – 04 – 2025	B	Arturo Quezada Daniel Rosado	Manual_de_usuario_2 .0.0
3.0.0	14 – 05 – 2025	A	Endrick Pool	Manual_de_usuario_3 .0.0

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

1.3 Histórico de cambios

Versión	Fecha	Cambios
1.0.1	26 – 02 – 2025	Se crea la estructura inicial del documento y las secciones de introducción, alcance, requisitos del sistema, instalación y ejecución.
1.0.2	27 – 02 – 2025	Se crean las secciones de uso básico y manejo de errores, también se hacen correcciones de formato.

2.0.0	02 – 04 – 2025	Se adapta la redacción del documento para hacer mención a que ahora se realiza el conteo de métodos por clase y se eliminan las referencias al conteo de líneas lógicas. Se cambia el punto 4 de la sección 8.3 “Uso básico del software” para que se alinee con la nueva salida esperada y también se modifica la imagen para reflejar la salida correspondiente.
3.0.0	14 – 05 – 2025	Se adapta el contenido y la redacción del documento para hacer referencia a las nuevas características de la aplicación, como la comparación de 2 proyectos de software.

2 CONTENIDO

1	Control de documentación	1
1.1	Control de configuración	1
1.2	Histórico de versiones	1
1.3	Histórico de cambios	1
2	Contenido	3
3	Introducción	4
4	Alcance	4
5	Referencias	4
6	Restricciones	4
7	Requisitos del Sistema	5
8	Instalación y ejecución	5
8.1	Ejecución desde el código fuente	5
8.2	Desde el archivo .exe	6
8.3	Uso básico del software	6
9	Manejo de errores y solución de problemas	9

3 INTRODUCCIÓN

ContadorAzul es una aplicación diseñada para agilizar el proceso de aplicación de la métrica de líneas de código, la cual abarca el aspecto de líneas físicas y métodos por clases. Así como la comparación de dos versiones distintas de un proyecto de software. Esta es aplicable únicamente para proyectos o programas realizadas bajo el lenguaje de programación de Java.

Los criterios para contabilizar las líneas físicas y métodos se han establecido en un estándar de Conteo, para el cual también se deberá de seguir lineamientos asociados al estándar de codificación recomendado.

El propósito de este manual de usuario es que los usuarios finales sepan cómo utilizar el programa de manera correcta; así como para que el equipo de desarrollo de este software de igual manera tenga la información necesaria para manejar el proyecto y darle su respectivo mantenimiento.

4 ALCANCE

En el presente documento se definen los aspectos que un usuario que desee hacer uso del sistema debe de considerar para que pueda realizar la ejecución del programa. Así mismo este ha de ser un guía para que el usuario pueda hacer uso del sistema, desde aspectos que van desde cómo realizar la ejecución del programa, así como su uso básico para poder implementar la métrica descrita. Así mismo se han de enumerar los casos para los cuales el programa no ha de funcionar de forma correcta y que el usuario no lo perciba como un fallo del programa.

5 REFERENCIAS

- Estándar de codificación
- Estándar de conteo

6 RESTRICCIONES

El sistema ha de contabilizar las líneas de código en base a las líneas físicas, así como los métodos contenidos dentro de una clase, por ende, si desea entender los criterios de conteo se ha de consultar el estándar de conteo definido.

Por otro lado, la funcionalidad de comparación de proyectos del sistema se basa en lo estipulado por el estándar de conteo el cual define los distintos tipos de líneas modificadas

que deben considerarse durante el análisis. Entre estos tipos se incluyen: Originales, modificadas, eliminadas, nuevas, cortada. Siguiendo los criterios específicos descritos en dicho estándar.

Así mismo, es importante que siga aspectos de formato definidos en el estándar de codificación que influyen en el conteo de líneas, por ende, se recomienda la revisión de los apartados asociados a:

- Bloques no vacíos: Style K&R
- No realizar declaraciones múltiples por línea
- Una anotación no esté en una sola línea ni bien formada.
- Las importaciones de paquete no sean explícitas ni contengan comodín *.

7 REQUISITOS DEL SISTEMA

Para poder hacer uso del sistema, deberá contar o instalar los siguientes aspectos:

- Sistema Operativo: Windows, MacOS
- Java: Versión 8 o superior.
- JDK: 21
- Maven: Versión 3.9.9 o superior

8 INSTALACIÓN Y EJECUCIÓN

8.1 Ejecución desde el código fuente

Para poder realizar la ejecución del código desde la terminal, deberá de seguir los siguientes pasos, así como aplicar los comandos indicados.

1. Abrir su consola de comandos
2. Navegar a la ruta raíz del proyecto a través del uso del comando `cd` seguido de la ruta del proyecto, por ejemplo: `cd /ruta/del/proyecto`
3. Se deberá de realizar el compilado del programa aplicando el siguiente comando:
`mvn clean package`
4. Posteriormente, se deberá de utilizar el siguiente comando para comenzar a ejecutar el archivo: *`java -jar target/coudecouter-1.0.0.jar`*
5. Deberá de ver la terminal con la ejecución del programa

```
PS C:\Users\cristianpan\Desktop\code-counter> java -jar target/codecounter-1.0.0.jar
Ingresa la ruta de la carpeta del proyecto:
```

Imagen 1. Ejecución en consola desde terminal

6. Continúe con el flujo descrito en el apartado 7.3

Nota: Para realizar esta ejecución deberá de contar con Maven y java, de cumplir con ambos requisitos deberá de abrir una consola de comandos y ejecutar los siguientes comandos:

- `java -version`
- `mvn -version`

Si visualiza la información podrá continuar con la ejecución, en caso contrario, deberá de instalar ambas dependencias.

8.2 Desde el archivo .exe

Este apartado considera que ya dispone del .exe, es decir, que no necesita descomprimirlo desde un .zip o su similar. En caso de ser así, descomprima el ejecutable y continúe.

1. Acceda a la ubicación del .exe desde su gestor de archivos
2. Realice doble clic, deberá de ver una terminal con un mensaje solicitando la ruta de su carpeta de proyecto.



Imagen 2. Ejecución del programa desde el ejecutable .exe

3. Continúe con el flujo descrito en el apartado 7.3

8.3 Uso básico del software

1. Tras haber iniciado la ejecución del programa, deberá de ver un mensaje solicitando la ruta del proyecto por analizar

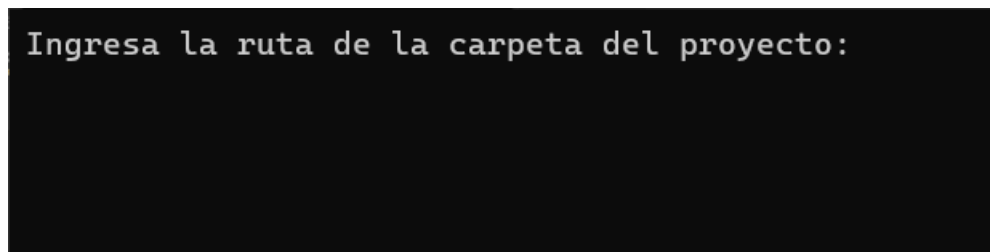


Imagen 3. Mensaje de solicitud de carpeta del proyecto

2. En este paso se decide que función del software desea utilizar, el análisis de un solo proyecto o la comparación de la versión de dos proyectos distintos.

3. *Analizar un solo proyecto:*

3.1. Introduzca a través del teclado la ruta del proyecto. Introduzca la ruta sin espacios en ella.

3.2. Si ya ha introducido la ruta de su proyecto, confirme la entrada a través de un Enter

3.3. Deberá de visualizar el resultado del conteo de métodos por clase, de líneas físicas por clase y del total de líneas físicas de las clases del código:

```
-----
Programa:
-----
```

Clases	Métodos	Lineas
JavaRegextConstants	1	11
ReasonInvalidFormat	3	19
FileNotFoundException	1	6
FolderNotFoundException	1	7
InvalidFormatException	4	15
JavaFilesNotFoundException	1	7
Main	2	13
CodeAnalyzer	13	102
StructCounter	9	53
JavaClass	8	47
JavaFile	5	29
ProgramBuilder	5	34
Reporter	1	9
TerminalReporter	4	38
FormatValidator	2	13
LogicalValidator	0	3
JavaFilesScanner	3	19
ImportValidator	3	20
SingleAnnotationValidator	3	18
SingleDeclarationValidator	2	14
StyleKAndRValidator	0	4
MethodDeclarationValidator	2	18
TypeDeclarationValidator	3	21
ValidatorManager	3	29
CodeAnalyzerTest	10	82
JavaFileTest	6	58
ImportValidatorTest	3	22
SingleAnnotationValidatorTest	5	33
SingleDeclarationValidatorTest	3	32
StyleKAndRValidatorTest	4	31
MethodDeclarationValidatorTest	8	60
TypeDeclarationValidatorTest	9	55

```
-----
Total de líneas físicas de código: 922
Presiona Enter para salir...
```

Imagen 4. Reporte de resultados de la aplicación de la métrica

3.7. Para cerrar el programa deberá de presionar Enter

4. *Comparar dos versiones de un proyecto:*

4.1. Introduzca a través del teclado la ruta de ambos proyectos separados por un espacio en el siguiente orden: "Versión_anterior Versión_actual". Las rutas no deben tener espacios.


```
C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\git\LinesStructCounterAzul>java -jar target\coudecounter-1.0.0.jar
Ingresa la ruta de la carpeta del proyecto:
C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\LinesStructCounter
```

Imagen 5. Entrada esperada para la comparación de proyectos

4.2. Si ya ha introducido ambas rutas de los proyectos, confirme la entrada a través de un Enter

4.3. Deberá de visualizar el resultado del conteo de métodos por clase, de líneas físicas por clase y del total de líneas físicas de las clases del código por cada proyecto, primero el de la versión anterior y luego el de la versión actual:

Programa: LinesStructCounter			Programa: LinesStructCounter_R		
Clases	Métodos	Líneas	Clases	Métodos	Líneas
JavaRegextConstants	1	11	JavaRegextConstants	1	11
ReasonInvalidFormat	3	19	ReasonInvalidFormat	3	19
FileNotFoundException	1	6	FileNotFoundException	1	6
FolderNotFoundException	1	7	FolderNotFoundException	1	7
InvalidFormatException	4	15	InvalidFormatException	4	15
JavaFilesNotFoundException	1	7	JavaFilesNotFoundException	1	7
Main	2	13	Main	2	13
CodeAnalyzer	13	102	CodeAnalyzer	13	102
ComparisonReport	3	28	JavaFileComparator	7	50
JavaFileComparator	9	71	ProjectComparator	5	53
ProjectComparator	5	48	STATUS	0	3
STATUS	0	3	StructCounter	9	53
StructCounter	9	53	JavaClass	8	47
JavaClass	8	47	JavaFile	5	29
JavaFile	5	29	ProgramBuilder	9	63
ProgramBuilder	9	70	Project	6	32
Project	6	32	ComparisonReport	7	46
Reporter	1	9	Reporter	1	9
TerminalReporter	4	38	TerminalReporter	4	38
FormatValidator	2	13	TxtReporter	2	25
LogicalValidator	0	3	FormatValidator	2	13
JavaFilesScanner	3	19	LogicalValidator	0	3
ImportValidator	3	20	JavaFilesScanner	3	19
SingleAnnotationValidator	3	18	ImportValidator	3	20
SingleDeclarationValidator	2	14	SingleAnnotationValidator	3	18
StyleAndRValidator	0	4	SingleDeclarationValidator	2	14
MethodDeclarationValidator	2	18	StyleAndRValidator	0	4
TypeDeclarationValidator	3	21	MethodDeclarationValidator	2	18
ValidatorManager	3	29	TypeDeclarationValidator	3	21
CodeAnalyzerTest	10	82	ValidatorManager	3	29
JavaFileTest	6	58	CodeAnalyzerTest	10	82
ImportValidatorTest	3	22	JavaFileTest	6	58
SingleAnnotationValidatorTest	5	33	ImportValidatorTest	3	22
SingleDeclarationValidatorTest	3	32	SingleAnnotationValidatorTest	5	33
StyleAndRValidatorTest	4	31	SingleDeclarationValidatorTest	3	32
MethodDeclarationValidatorTest	8	60	StyleAndRValidatorTest	4	31
TypeDeclarationValidatorTest	9	55	MethodDeclarationValidatorTest	8	60
Total de líneas físicas de código: 1140			TypeDeclarationValidatorTest	9	55
			Total de líneas físicas de código: 1160		

Imagen 6. Reportes de resultados de la aplicación de la métrica por cada proyecto

4.7. Tras haber realizado la comparación de los proyectos, deberá de ver un mensaje solicitando la ruta de guardado de los reportes de comparación

```
Ingresa la ruta del directorio para guardar los reportes:
C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes
```

Imagen 7. Mensaje de solicitud de carpeta de guardado de los reportes

4.8. Introduzca a través del teclado la ruta en la que desea guardar los reportes

4.9 Si ya ha introducido la ruta de guardado, confirme la entrada a través de un Enter

4.10. Deberá de visualizar el resultado haber generado los reportes por cada clase, así como un resumen general de los cambios entre las versiones:

```
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\TypeDeclarationValidatorTest.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\CodeAnalyzerTest.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\TypeDeclarationValidator.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\StyleKAndRValidator.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\TxtReporter.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\LineRecord.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\Reporter.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\JavaClass.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\ProjectComparator.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\JavaRegextConstants.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\LogicalValidator.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\ReasonInvalidFormat.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\ImportValidatorTest.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\StyleKAndRValidatorTest.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\ProgramBuilder.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\SingleDeclarationValidator.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\FolderNotFoundException.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\SingleAnnotationValidator.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\ValidatorManager.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\JavaFilesScanner.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\InvalidFormatException.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\TerminalReporter.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\SingleDeclarationValidatorTest.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\FileNotFoundException.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\ComparisonReport.java_report.txt
Reporte generado: C:\Users\EPoLF\OneDrive\Escritorio\Proyecto_3\reportes\CodeAnalyzer.java_report.txt

=== RESUMEN GLOBAL DE CAMBIOS ===
Total de cambios en todo el proyecto:
Nuevas líneas (NEW): 403
Líneas modificadas (MODIFIED): 0
Líneas eliminadas (DELETED): 350
Líneas sin cambios (ORIGINAL): 3424

Total de líneas procesadas: 4178

Presiona Enter para salir...
|
```

Imagen 8. Mensaje de solicitud de carpeta de guardado de los reportes

4.11 Para cerrar el programa deberá de presionar Enter

9 MANEJO DE ERRORES Y SOLUCIÓN DE PROBLEMAS

Los errores obtenidos durante la ejecución del programa serán mostrados a través de la terminal, de tal manera que interrumpirá la ejecución del programa y deberá de ser reiniciado.

Los errores que han de surgir serán asociados al incumplimiento de las restricciones asociadas al estándar de codificación mencionadas en el apartado correspondiente.