

La Tabla 5 presenta las métricas de legibilidad compuestas más importantes.

Tabla 5 – Listado de métricas de legibilidad compuestas

Métrica	Descripción	Uso tradicional	
ARI : Índice de legibilidad automático	Se basa en la relación entre la dificultad de la oración y la dificultad de la palabra. La dificultad de las oraciones se determina como palabras por oración, y la dificultad de las palabras es al calcular letras por palabras. Ecuación: $ARI = 4.71 (\text{caracteres}) + 0.5 (\text{palabras}) - 21.43$. El valor de ARI significa el nivel de grado escolar necesario para que una persona comprenda ese texto.	Legibilidad texto	del texto
SMOG: Medida simple de Gobbledygook	Autor: G Harry McLaughlin [22] en 1969. SMOG representa el nivel de grado escolar necesario, para que una persona pueda leer ese texto. Ecuación: $SMOG = 3 + \text{Raíz cuadrada de recuento polisillable (conteo de palabras)}$	Legibilidad texto	del texto
FKI: Índice de legibilidad de Flesch-Kincaid	Ecuación: $FKI = 206.835 - (1.015 * (\text{palabras totales} / \text{sentencias totales})) - 84.6 * (\text{silabas totales} / \text{palabras totales})$. Un alto FKI indica una alta legibilidad del código, y un valor bajo que el código es difícil de leer. El rango es de 0 a 100.	Legibilidad texto	del texto
GFI: índice de confusión de Gunning (FOG)	Autor: Robert Gunning. Utiliza la longitud promedio de las oraciones y el porcentaje de palabras difíciles. Ecuación: $GFI = 0.4 (ASL + PHW)$ donde, ASL = longitud promedio de la oración y PHW = porcentaje de palabras difíciles. Un puntaje GFI de 5 significa legible, 10 significa duro, 15 significa difícil y 20 indica muy difícil.	Legibilidad texto	del texto
CLI: Índice Coleman-Liau.	Autores: Meri Coleman y T. L. Liau. Ecuación: $CLI = 0.0588L - 0.296S - 15.8$, donde, L = número promedio de letras, S = número promedio de sentencias, CLI representa el nivel de grado escolar necesario, para que una persona pueda leer adecuadamente ese texto.	Legibilidad texto	del texto
SRES: fácil puntaje de legibilidad de software	Autores: Abbas, Borestler, Cspersend, and Nordstrin (2009). Ecuación: $SRES = ASL - 0.1 AWL$. Dónde, ASL es la métrica de longitud promedio de sentencia, que es el número promedio de tokens por declaración, y AWL es la métrica de longitud promedio de palabra. A mayor SRES, más difícil para leer.	Legibilidad código fuente	del código fuente
PHD: Puntuación de legibilidad de Posnett	Autores: Posnett et al. (2011). Ecuación: $PHD = 1 / (1 + e^Z)$ Donde, $Z = 8.87 - 0.033 V + 0.40 LOC - 1.5 H$, Donde, V es el volumen de Halstead, LOC las líneas de código y H es la información Shannon de tokens, cuyas fórmulas son: $V = N \log_e n$; $H(s) = \sum_{i=1}^n p_i \log p_i$. Donde, p_i es la fracción de tokens (s_i) en una fuente S	Legibilidad código fuente	del código fuente
Legibilidad cuantitativa optimizada de Choi	Ecuación: $\text{Legibilidad} = (-0.020) * (LOC) + (0.040) * (\# \text{ de comentarios}) + (0.037) * (\# \text{ de líneas en blanco}) + (-0.755) * (\# \text{ de operadores bit a bit}) + (-0.153) * (\text{control anidado máximo}) + (-0.001) * (\text{Volumen de programa}) + (-0.611) * (\text{Entropía})$	Legibilidad código fuente	del código fuente
WSCR: legibilidad centrada en servicio web	WSCR determina el puntaje de legibilidad para un documento WSDL. Considera las características: alcance del documento, cohesión del documento y el índice simplificado de Dale-Chall. $WSCR(d) = \text{Alcance}(d) + \text{Cohesión}(d) + \text{DaCw}(d)$. Utiliza términos existentes del diccionario de WordNet, como entrada para calcular el valor de legibilidad.	Legibilidad texto	del texto
AR: promedio de legibilidad; RiC: legibilidad en la clase; RiG: en el paquete.	Autor: Liu et al. (2015) Ecuaciones: $AR = \text{Comentarios válidos} / \text{Todos los resultados} $; $RiC = \text{comentarios válidos en la clase} / \text{Todos los resultados en la clase} $; $RiG = \text{sumatoria de todos los RiC del paquete}$; Si RiC o $RiG < AR$, entonces la legibilidad es baja. Si RiC y $RiG > AR$, entonces es legible.	Legibilidad código fuente	del código fuente
CIC: Consistencia de los comentarios e identificadores.	Autor: Scalabrino et al. (2018). Ecuación: $CIC(m) = \text{comentarios}(m) \cap \text{identificadores}(m) / \text{comentarios}(m) \cup \text{identificadores}(m) $ donde, m es el método. Teniendo en cuenta que pueden existir sinónimos de palabras en comentarios e identificadores, se tiene una variación de la ecuación: $CIC(m)_{sin} = \text{comentarios}(m) \cap \text{identificadores}(m) \cup \text{sinónimos}(m) / \text{comentarios}(m) \cup \text{identificadores}(m) \cup \text{Sinónimos}(m) $ donde, sinónimos: conjunto de sinónimos de los términos que están en los identificadores.	Legibilidad código fuente	del código fuente
ITID: Términos id en el diccionario.	Autor: Scalabrino et al. (2018). Fue inspirado por las características de lenguaje natural (lingüísticas) introducidas por Dorn. $ITID(I) = \text{Terminos}(I) \cap \text{Diccionario} / \text{Terminos}(I) $ donde, Términos: es el conjunto de términos extraídos de una línea I de un método. Diccionario: es un conjunto de palabras en un diccionario.	Legibilidad código fuente	del código fuente
NMI: Identificadores de significado estrecho.	Autor: Scalabrino et al. (2018). Los términos deben tener un significado específico, no genérico que pueda generar confusiones. $NMI(I) = \sum_{t \in C_I} \text{particularidad}(t)$ donde, t: es un término extraído de la línea de código I. Particularidad(t): es calculado como el número de saltos desde el nodo t que lo contiene hasta el nodo raíz del árbol hypernym de t. Usan el árbol hypernym/hyponym del lenguaje inglés definido para WordNet.	Legibilidad código fuente	del código fuente
CR: Legibilidad de los comentarios.	Autor: Scalabrino et al. (2018). Calcula la legibilidad de los comentarios (CR) usando el índice Flesch-Kincaid comúnmente utilizado para evaluar la legibilidad de textos en lenguaje natural. El índice considera 3 tipos de elementos: palabras, sílabas y frases. Ecuación: $FK(S) = 206.835 - 1.015 * (\text{palabras}(S) / \text{frases}(S)) - 84.600 * (\text{sílabas}(S) / \text{Palabras}(S))$ donde, S: es el fragmento de código fuente. Se calcula el CR al (1) juntar todas las líneas comentadas del fragmento S; (2) unir los comentarios con un carácter " ", para asegurar de que no se unan diferentes comentarios creando una sola frase; y (3) calcular el índice de Flesch-Kincaid en dicho texto.	Legibilidad código fuente	del código fuente
NM: número de significados.	Autor: Scalabrino et al. (2018). El significado de una palabra puede ser polisémico (tener más de un significado). La característica mide el número de significados (NM) o el nivel o polisemia de un fragmento. Miden el número de significados derivados de WordNet.	Legibilidad código fuente	del código fuente
TC: Coherencia textual.	Autor: Scalabrino et al. (2018). Se analiza el código fuente y se construye el árbol de sintaxis abstracta (AST) para detectar bloques sintácticos. Se calcula la superposición de vocabulario entre todos los posibles pares de bloques sintácticos distintos. La coherencia textual (TC) de un fragmento se puede calcular como el máximo, el mínimo o la superposición promedio entre cada par de bloques sintácticos.	Legibilidad código fuente	del código fuente
NOC: Número de conceptos.	Autor: Scalabrino et al. (2018). Captura directamente el número de conceptos implementados en un fragmento de código a nivel de línea. Crea un documento por cada línea. Utiliza una técnica de clustering basada en densidad, DBSCAN. Miden la distancia entre 2 documentos así: $NOC_{dist}(d_1, d_2) = d_1 \cap d_2 / d_1 \cup d_2 $ número de conceptos de un fragmento m: $NOC(m) = \text{Clusters}(m) $ Normalizando $NOC: NOC_{norm}(m) = \text{Clusters}(m) / \text{Documentos}(m) $	Legibilidad código fuente	del código fuente
Legibilidad de la variable.	Autor: Xu et al. (2017). La legibilidad de una variable "v" en un método "m", legibilidad (v, m), está definida como: $\text{Legibilidad}(v, m) = \text{Concreción}(v)$	Legibilidad código fuente	del código fuente
Legibilidad de un método.	Autor: Xu et al. (2017). La legibilidad de un método o constructor "m" en una clase "c" está definida como la legibilidad promedio de todas las variables "V" usadas en "m", incluyendo variables locales, parámetros, constantes nombradas, variables instanciadas y variables de clase. $\text{Legibilidad}(m, c) = \sum_{i=1}^n \text{Legibilidad}(vi, m) / V $	Legibilidad código fuente	del código fuente
Legibilidad de la clase.	Autor: Xu et al. (2017). La legibilidad de la clase C en un programa orientado a objetos P es definido como la legibilidad promedio de todos los métodos y constructores en "C". $\text{Legibilidad}(c, P) = \sum_{i=1}^n (\text{Legibilidad}(mi, c) / M)$	Legibilidad código fuente	del código fuente
Legibilidad del programa.	Autor: Xu et al. (2017). La legibilidad de un programa orientado a objetos P está definida como la legibilidad promedio de todas las clases en P. $\text{Legibilidad}(P) = \sum_{i=1}^n R(ci, P) / C $	Legibilidad código fuente	del código fuente