

Característica	Sistema de Logging (Logback, Log4j)	System.out.println
Control de Nivells	Sí. Pots classificar els missatges (ERROR, WARN, INFO, DEBUG). Pots decidir mostrar només els errors en producció i tot el detall en desenvolupament, sense canviar el codi.	No. O s'imprimeix tot o no s'imprimeix res. Per treure missatges, has d'esborrar el codi i recompilar.
Rendiment	Eficient. La construcció del missatge només es fa si el nivell de log està activat. Per exemple, <code>logger.debug("Dada: " + obj.toString())</code> no executa <code>toString()</code> si el nivell DEBUG està apagat.	Ineficient. La concatenació de strings i les crides a mètodes es fan sempre, malgastant recursos encara que no necessitis veure la sortida.
Context Automàtic	Sí. Afegeix automàticament informació crucial com la data i hora, el nom de la classe, el fil d'execució (thread) i el nivell del missatge.	No. Només imprimeix el text que li passes. Si vols la data, l'has d'afegir manualment a cada línia.
Configuració de la Sortida	Molt flexible. Pots enviar els logs a múltiples destins alhora: la consola, fitxers de text (amb rotació automàtica), bases de dades, serveis externs o fins i tot enviar un email en cas d'error crític.	Rígid. Només escriu a la sortida estàndard de la consola (stdout).
Format i Llegibilitat	Personalitzable. Pots definir un patró per a tots els missatges (amb colors, tabulacions, etc.), fent que els logs siguin estructurats i fàcils de llegir i filtrar.	Sense format. És un flux de text desestructurat, on és difícil distingir un error crític d'un simple missatge informatiu.