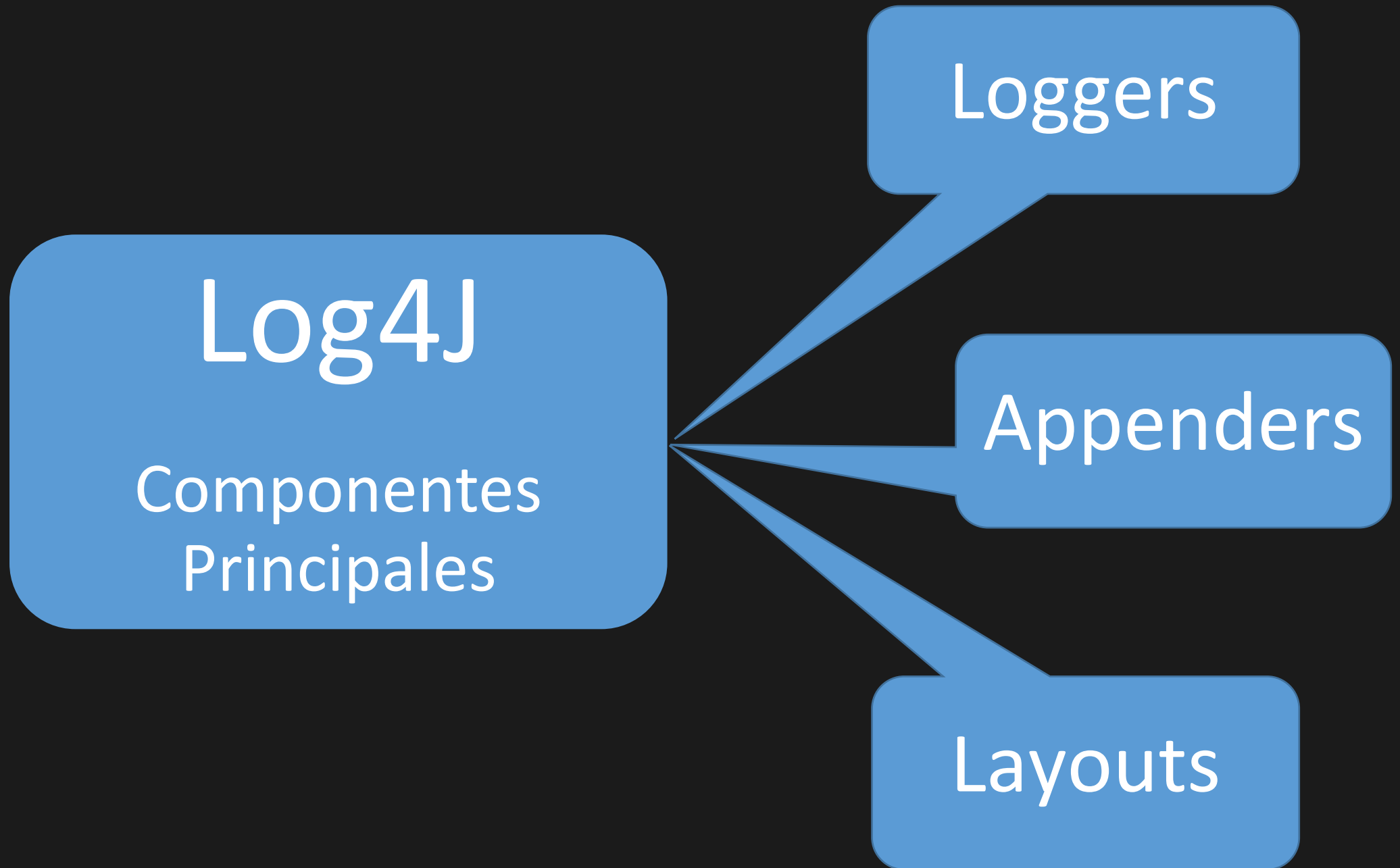


LOG4J



¿Qué es Log4J?

- Biblioteca de gestión de registros
- Gestión flexible de registros
- Personalización de configuración
- Niveles de registros claros
- Integración sencilla
- Aplicaciones prácticas



Configuración

1. Importar biblioteca:



MVN Repository: Log4J API



MVN Repository: Log4J Core

2. Creación del archivo de configuración:

log4j2.properties / log4j2.xml

3. Niveles de registro:

DEBUG, INFO, WARN, ERROR, FATAL

Configuración

4. Configuración de appenders:
 Consola, archivo, base de datos, etc...
5. Personalización del formato de salida (opcional)

Layout

PatternLayout: patrón de formato utilizando ciertas cadenas de caracteres especiales que representan diferentes partes del mensaje de registro, como la fecha, el nivel de registro, el nombre del logger, etc.

SimpleLayout: formato simple, mostrando solo el nivel de registro y el mensaje.

HTMLLayout: formateados en HTML. Es útil cuando se necesita generar registros para visualización en un navegador web.

XMLLayout: formato XML.

JSONLayout formato JSON.

PatternSelector: permite seleccionar un layout específico basado en un patrón definido. Por ejemplo, se puede configurar para que use HTMLLayout para ciertos tipos de registros y PatternLayout para otros.

SyslogLayout: Este layout genera registros formateados de acuerdo con la especificación del protocolo syslog, que es comúnmente utilizado en sistemas Unix y Linux.

PatternLayout

`%m`: El mensaje del registro.

`%p`: El nivel de prioridad del registro (DEBUG, INFO, WARN, ERROR, FATAL).

`%r`: Tiempo en milisegundos desde que se inició el programa hasta que se registró el mensaje.

`%c`: El nombre de la Clase.

`%t`: El nombre del thread que registró el mensaje.

`%d{formato}`: La fecha y hora del registro, donde el formato puede ser especificado según el formato de `SimpleDateFormat` en Java.

`%F`: El nombre del archivo donde se registró el mensaje.

`%L`: El número de línea en el archivo donde se registró el mensaje.

`%n`: El separador de línea (salto de línea).

`%X`: Muestra el valor de un campo en el `Mapped Diagnostic Context` (MDC).

`%x`: Muestra el valor de un campo en el `Nested Diagnostic Context` (NDC).

%M: El nombre del método que generó el mensaje de registro.

%throwable: Muestra el stack trace de una excepción si está disponible en el registro.