Log4J

Parte A)

1) La etiqueta <Loggers> se usa para definir los distintos "canales" de logueo que queremos configurar en nuestra aplicación.

2) La etiqueta <Appenders> sirve para definir los destinos de los logs, o sea, dónde se van a guardar o mostrar los mensajes que genera la app.

3) El Root Logger es el logger principal o base, y se configura dentro de la etiqueta <Root> que va dentro de <Loggers>.

Tiene dos funciones principales:

- Definir el nivel mínimo de log que se va a aceptar (por ejemplo, si ponés level="info", no se mostrarán los DEBUG).
- Asignar Appenders por defecto, que van a recibir todos los logs que no tengan un logger específico.

4) Apache Flume es una herramienta de recolección y transporte de grandes cantidades de datos de logs o eventos, de forma eficiente y escalable.

Se usa sobre todo para enviar datos (como logs de servidores web) hacia sistemas como Hadoop o bases de datos, para después analizarlos.

Parte B)

4) Este appender imprime el log en formato HTML. Es decir, en vez de líneas simples, te muestra los mensajes como si fueran parte de una tabla HTML (útil para ver logs en un navegador).

```
<!DOCTYPE html>
<html>
 <head><title>Log</title></head>
 <body>
  <table>
   <tr><th>Time</th><th>Level</th><th>Logger</th><th>Message</th></tr>
   <tr><td>12:34:56</td><td>INFO</td><td>Milog</td><td>La app se inició</td></tr>
  </table>
 </body>
</html>
```

5) <Configuration status="WARN">
 <Appenders>
  <RollingFile name="FileLogger" fileName="logs/app.log"
        filePattern="logs/app-%d{yyyy-MM-dd-HH-mm}.log">
   <PatternLayout pattern="%d [%t] %-5level %logger{36} - %msg%n"/>
   <Policies>
    <TimeBasedTriggeringPolicy interval="2" modulate="true"/>
   </Policies>
  </RollingFile>
 </Appenders>

 <Loggers>
  <Root level="info">
   <AppenderRef ref="FileLogger"/>
  </Root>
 </Loggers>
</Configuration>

6) <Configuration status="WARN">
 <Appenders>
  <RollingFile name="FileLogger" fileName="logs/cron.log"
        filePattern="logs/cron-%d{yyyy-MM-dd-HH-mm}.log">
   <PatternLayout pattern="%d [%t] %-5level %logger{36} - %msg%n"/>
   <Policies>
    <CronTriggeringPolicy schedule="0 0/2 * * * ?" />
   </Policies>
  </RollingFile>
 </Appenders>

 <Loggers>
  <Root level="info">
   <AppenderRef ref="FileLogger"/>
  </Root>
 </Loggers>
</Configuration>

7) <Appenders>
 <!-- Formato 1 -->
 <Console name="SimpleConsole">
  <PatternLayout pattern="%d{HH:mm:ss} [%p] %m%n"/>
 </Console>

 <!-- Formato 2 -->
 <Console name="ThreadConsole">
  <PatternLayout pattern="[%t] %c - %m%n"/>
 </Console>

 <!-- Formato 3 -->
 <Console name="DetailedConsole">
  <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss} %-5p [%t] %c{1} - %m%n"/>
 </Console>

 <!-- Formato 4 -->
 <Console name="DevConsole">
  <PatternLayout pattern="%d [%p] (%F:%L) %m%n"/>
 </Console>
</Appenders>

8) a. %d [%-6p] %c{1} - %m%n
%d → Fecha/hora
%-6p → Nivel (INFO, ERROR, etc.) alineado a 6 caracteres
%c{1} → Nombre corto del logger (último fragmento del package)
%m → El mensaje
%n → Salto de línea
b. %sn %d{yyyy/MM/dd HH:mm:ss,SSS} %r [%-6p] [%t] %c{3} %C{3}.%M(%F:%L) - %m%n
Fecha detallada
%r → Tiempo desde que empezó el programa (en ms)
Nivel
Thread
Logger (últimos 3 paquetes)
Clase, método, archivo y línea
Mensaje

9)ThreadContext permite agregar información adicional por hilo (como usuario, IP, módulo, etc.), que podés incluir en tus logs.