

# Monitorización de aplicaciones en Java

## VisaulVM

<https://visualvm.github.io/> Lo primero que necesitamos es descargar e instalar la aplicación en nuestro equipo cliente. Una vez instalada si queremos monitorizar un tomcat ubicado en otro equipo necesitamos definir ,en dicho tomcat, una serie variables para poder conectarnos y monitorizar su estado.

Editamos el fichero /opt/tomcat/bin/catalina.sh y añadimos/creamos la siguiente variable CATALINA\_OPTS con las siguientes líneas:

```
CATALINA_OPTS="-Dcom.sun.management.jmxremote=true -  
Dcom.sun.management.jmxremote.port=9090 -  
Dcom.sun.management.jmxremote.ssl=false -  
Dcom.sun.management.jmxremote.authenticate=false"
```

Reiniciamos el tomcat. Abrimos la aplicación visualvm y añadimos una nueva conexión a la ip del tomcat por el puerto 9090

## JMeter

[Apache JMeter](#) es una herramienta de prueba de carga para analizar y medir el rendimiento de nuestra aplicación web

- <https://www.adictosaltrabajo.com/tutoriales/jmeter-navegacion/>

## Psi-Probe

Esta es una aplicación web que desplegamos en nuestro servidor y con la cual podremos monitorizar el rendimiento de diversos parámetros.

La aplicación war. se puede desacargar del siguiente enlace

<https://github.com/psi-probe/psi-probe/releases>

## Referencias

- <https://tododev.wordpress.com/2012/02/16/entender-detectar-y-localizar-memory-leaks-en-aplicaciones-web-java-lang-outofmemoryerror/>
- <https://cmop17.wordpress.com/2016/04/11/tomcat-configurar-jmx-para-monitorear-recursos-y-deseempeno-del-servidor/>
- <https://www.genbetadev.com/herramientas/the-eclipse-memory-analyze-una-excelente-herramienta-para-buscar-memory-leak>
- <http://blog.markshead.com/1129/connecting-visual-vm-to-tomcat-7/>

From:

<http://intrusos.info/> - **LCWIKI**

Permanent link:

<http://intrusos.info/doku.php?id=aplicaciones:java:monitorizacion>

Last update: **2023/01/18 14:36**

