

5. Discusión

Tras la realización ya de todo el trabajo, se puede sacar una serie de conclusiones. En primer lugar, se realizará una discusión sobre los resultados obtenidos experimentalmente sobre el consumo de CPU de Project.net.

En la página web de Project.net vienen los requerimientos computacionales que debe soportar una máquina en la que se quiera instalar Project.net [4]. Suponiendo, que el número de usuarios simultáneos máximo sería de 200, los requerimientos serán los que se muestran en la siguiente tabla:

Servidor	CPU (min)	Memoria
Aplicación o BBDD	2,5 Ghz (dual core)	2 GB
Aplicación y BBDD	2,5 Ghz (dual core)	4 GB

Como se explicó en el apartado de Metodología, Project.net se instaló en una máquina física (Core Processor 2.71 Ghz y 2.87 GB de RAM), y en una máquina virtual (Intel Core i3 2.2 Ghz, 512 Megas de RAM). Vemos, que ninguna de las máquinas, cumple los requisitos. En la máquina virtual, tanto la capacidad del microprocesador como la memoria del sistema, se quedan cortas. Recordamos, que cuando se probó Tomcat en la máquina virtual con JProfiler, el uso de consumo CPU hasta que Tomcat se iniciaba completamente, llegaba hasta el 90 por ciento, y posteriormente, al navegar por Project.net, también se alcanzaba ese valor. Ocurría eso con un sólo usuario navegando por Project.net, imaginemos que existen más usuarios, y que, en la máquina también se está ejecutando otro tipo de software, un antivirus, por ejemplo, el comportamiento de la máquina servidor será totalmente impredecible, lo más seguro es que falle o el tiempo de respuesta de Tomcat sea bastante lento.

En cambio, Project.net en la máquina física, tiene unos resultados un poco más satisfactorios, el uso de CPU consumida no supera el 50 % en ningún momento. Sin embargo, ese valor sigue siendo demasiado alto. La recomendación de Project.net sobre la memoria RAM mínima (4 Gb) parece ser bastante acertada.

De cualquier manera, otra opción, si se quiere reducir el consumo de CPU en la máquina servidor, podría ser montar una arquitectura con tres capas, una de ellas para el contenido estático, otra para el dinámico, y una tercera para datos (BBDD). Habría que configurar en Tomcat un conector JK2 a un servidor web externo, por ejemplo, Apache, este servidor se usaría para servir el contenido estático, puesto que es más rápido que Tomcat al servir páginas estáticas. Tomcat se seguiría usando para servir contenido dinámico [14].

Si los tres servidores, Apache, Tomcat, y el servidor de la base de datos, se instalan en tres máquinas diferentes, los recursos que se consumirán en cada una de ellas serán menores, y consecuentemente, la eficiencia computacional de Project.net también será mayor [13]. La arquitectura de este sistema sería la que se representa en la siguiente figura.

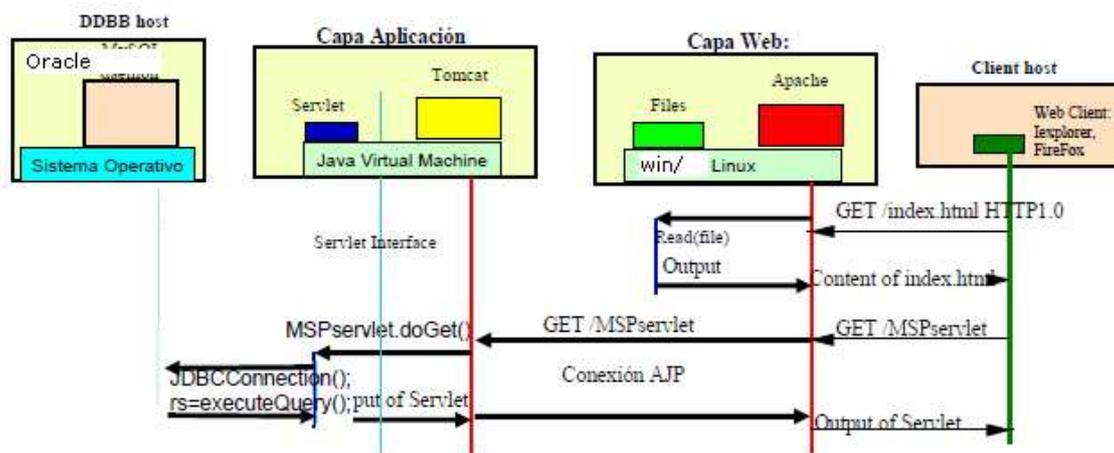


Ilustración 51: Arquitectura multicapa, extraída de [13]

En adelante, la discusión se enfocará al aspecto funcional de Project.net. Sobre esto, en primer lugar decir, que la implementación de Project.net es relativamente sencilla, y esto le atribuye una gran ventaja. En la página web del autor existe un foro donde ofrecen ayuda ante dificultades técnicas. De hecho, para llevar a cabo este Proyecto ha resultado bastante útil ese foro, por los post antiguos, y por las respuestas recibidas ante algunas preguntas planteadas. Una vez instalado, Project.net resulta bastante intuitivo, y su uso es muy fácil. Existe también en la web un manual completo de su funcionamiento (en inglés). Para gestionar tareas, la asignación de ésta a una persona o varias, visualizar de forma centralizada los proyectos en curso de una empresa, obtener diagramas de Gantt de forma automática, incentivar el trabajo colaborativo, llevar un control sobre la productividad de cada trabajador, o informar sobre el estado financiero del proyecto, resulta una herramienta excelente. Posee además la ventaja de que se trata de una herramienta de software libre, por lo que no existen costes en licencias. Hay que destacar también, que cualquier trabajador puede acceder a la plataforma desde cualquier parte del mundo, sólo con un acceso a internet, siempre que, el servidor esté funcionando correctamente. Sin embargo, Project.net carece de ciertas funcionalidades.

Project.net permite recopilar, almacenar y recuperar la información de un proyecto, pero carece de herramientas para realizar modelos de decisión para optimizar costes a partir de esos datos. Esto supone una desventaja importante para los usuarios de Project.net. En cuanto al tema financiero, Project.net sólo permite informar al resto de participantes de este estado, pero no se pueden asignar costes fijos a cada tarea ni recurso.

Otra carencia de Project.net es que carece de un método, como el PERT, que permite evaluar los riesgos, y en función de ellos, calcular una estimación más realista de la duración del proyecto, y sus consecuentes costes, si éste se prolonga durante un tiempo.

Todas esas carencias de Project.net, en la solución EPM de Microsoft están solventadas. ¿Qué conclusión podemos sacar entonces?, ¿cuál es mejor opción? La respuesta a esas preguntas depende de la utilidad o aplicación que se le pretenda dar a la herramienta.

Supongamos el caso de un grupo de investigación de una universidad. Lo prioritario sería poder crear un plan de trabajo, asignar tareas a personas con un plazo limitado, que éstas vayan informando de los avances logrados, poseer un blog o wikis para cualquier tipo de anuncio o si surge cualquier problema, haciendo posible el trabajo de forma colaborativa. En ese caso, las funcionalidades de Project.net serían suficientes.

Sin embargo, si estuviésemos en el caso de una gran empresa, donde la toma de decisiones y el hecho de que pueda producirse cualquier suceso que retrase la duración del proyecto es algo muy significativo, pues de ello depende que pueda producirse grandes ganancias o pérdidas económicas, las funcionalidades de Project.net se quedarían cortas.

Por tanto, se puede decir finalmente, que elegir Project.net como herramienta PPM para una pequeña o mediana empresa para gestionar y planificar los proyectos en curso, asignando tareas específicas a una o varias personas, controlando sus tiempos, es una decisión completamente acertada, puesto que las funcionalidades son suficientes, la licencia es gratuita, y la implantación y el posterior uso de Project.net es fácil.