

## **1. Introducción**

### **1.1. Antecedentes**

En la actualidad, existen proyectos cada vez más complejos, cuyas exigencias en tiempo y costes, exigen una administración profesional y centralizada para maximizar la probabilidad de éxito del proyecto. Este hecho hace que la mayoría de empresas y organizaciones, cuenten con soluciones centralizadas para Administración de Proyectos y Portafolios (PPM) [12].

Algunas de las ventajas de las herramientas PPM son:

- Administración efectiva de recursos y proyectos.
- Ayudar a implementar mejores prácticas a la hora de abordar cualquier proyecto.
- Compartir y administrar de forma centralizada a los recursos, de acuerdo a sus capacidades y disponibilidad.
- Permite la interacción. Los responsables pueden ver el estado de los proyectos, el administrador puede dar de alta actividades, asignar responsabilidades y tiempos, y el equipo de trabajo puede actualizar el avance de sus tareas.

El interés de llevar a cabo este Proyecto, tiene como origen el auge de las herramientas PPM.

Existe una larga lista de herramientas PPM, entre ellas se encuentra Project.net, y algunas tan conocidas, como Microsoft Project [10].

Algunas de las características a comparar entre estas herramientas, es el tipo de licencia, libre o privada, siendo éstas últimas las que más predominan, la arquitectura o su tecnología.

Project.net es de licencia libre, y hace uso de la tecnología JSP de Java, similar a ASP, usada por Microsoft. Readmine, otra herramienta PPM, con la que compararemos a Project.net en un apartado siguiente, se programa en Ruby.

Estas aplicaciones necesitarán alojar la información de cada uno de los proyectos en bases de datos, en el caso de Project.net, sólo es compatible con Oracle, suponiendo una clara desventaja, sobre todo, tratándose de una herramienta de software libre. En otras herramientas, se podrá trabajar también con MySQL, PostgreSQL o SQLITE.

Una vez descrito el contexto en el que se encuentra Project.net, se hará una breve introducción de Project.net.

## 1.2. Project.net

Project.net es una herramienta Project Portfolio Management (PPM) de uso libre, que se emplea para gestionar proyectos de forma colaborativa [1].

El trabajo colaborativo promueve a un grupo de personas a lograr metas comunes, objetivos específicos, haciendo uso de herramientas diseñadas para dar soporte y facilitar el trabajo.

En el marco de una organización, el trabajo con soporte tecnológico se presenta como un conjunto de estrategias que tienden a maximizar los resultados y minimizar la pérdida de tiempo.

Project.net puede considerarse como el entorno en el cual todos los participantes del proyecto trabajan, colaboran y se ayudan para la realización del proyecto.

En Project.net existe la figura de un administrador, que es el que puede invitar a otros miembros a unirse al proyecto, cada uno de los cuales, tiene un papel/rol en la empresa.

Project.net tiene una arquitectura de Cliente-Servidor.

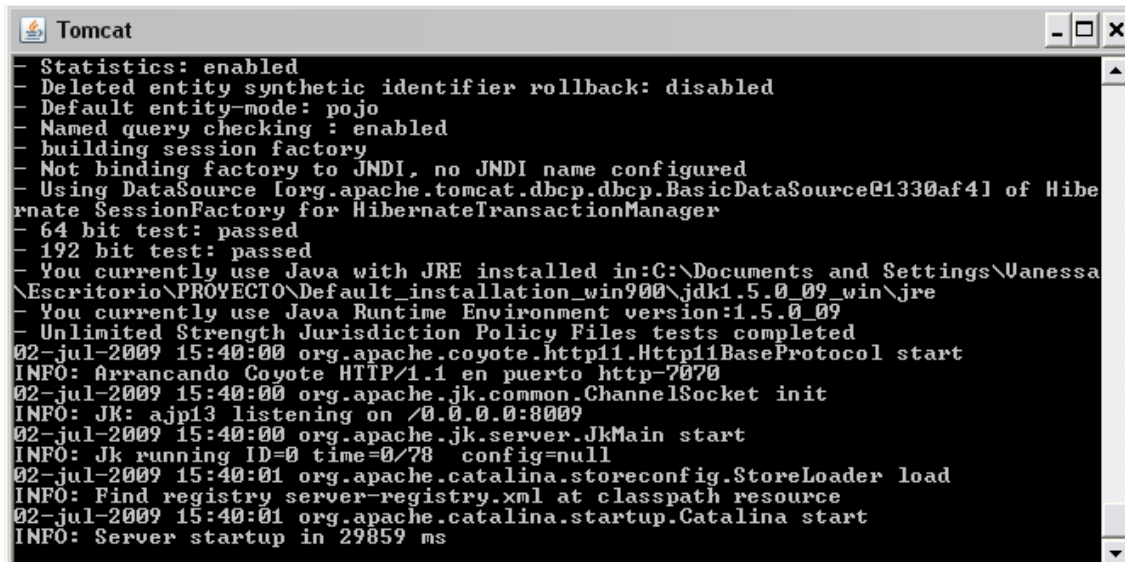
Existe un programa (servidor) que se está ejecutando continuamente en un ordenador, el servidor, mientras que, los usuarios, pueden acceder a Project.net desde Internet haciendo uso del protocolo HTTP.

El servidor está a la espera de peticiones HTTP por parte de un cliente HTTP, que conocemos como navegador. En el lado del cliente sólo se necesitará conexión a internet y un navegador [2].

La versión de Project.net en la que vamos a centrar nuestro estudio es en la 9.0. Esta versión requiere del servidor Apache Tomcat 5 para funcionar, funcionando éste como un contenedor de serles, siendo éstos, programas escritos en Java. Versiones anteriores de Project.net hacían uso también del servidor BEA Weblogic. Podemos decir que Project.net es una aplicación web que corre sobre Apache Tomcat 5.

Para entender cómo funciona Project.net, debemos comprender cómo funciona Apache Tomcat, su arquitectura y protocolos de comunicación [15].

Todo eso se desarrollara con detalle en el anexo.



```
Tomcat
- Statistics: enabled
- Deleted entity synthetic identifier rollback: disabled
- Default entity-mode: pojo
- Named query checking : enabled
- building session factory
- Not binding factory to JNDI, no JNDI name configured
- Using DataSource org.apache.tomcat.dbcp.dbcp.BasicDataSource@1330af41 of Hibernate SessionFactory for HibernateTransactionManager
- 64 bit test: passed
- 192 bit test: passed
- You currently use Java with JRE installed in:C:\Documents and Settings\Uanessa
\Escritorio\PROYECTO\Default_installation_win900\jdk1.5.0_09_win\jre
- You currently use Java Runtime Environment version:1.5.0_09
- Unlimited Strength Jurisdiction Policy Files tests completed
02-jul-2009 15:40:00 org.apache.coyote.http11.Http11BaseProtocol start
INFO: Arrancando Coyote HTTP/1.1 en puerto http-7070
02-jul-2009 15:40:00 org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
02-jul-2009 15:40:00 org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/78 config=null
02-jul-2009 15:40:01 org.apache.catalina.storeconfig.StoreLoader load
INFO: Find registry server-registry.xml at classpath resource
02-jul-2009 15:40:01 org.apache.catalina.startup.Catalina start
INFO: Server startup in 29859 ms
```

Ilustración 1: Apache Tomcat a la espera de solicitudes HTTP

En la figura anterior podemos ver, el servidor esperando peticiones HTTP. Mientras que en la figura siguiente, vemos la página de inicio que le aparece a un usuario cuando intenta acceder a Project.net.

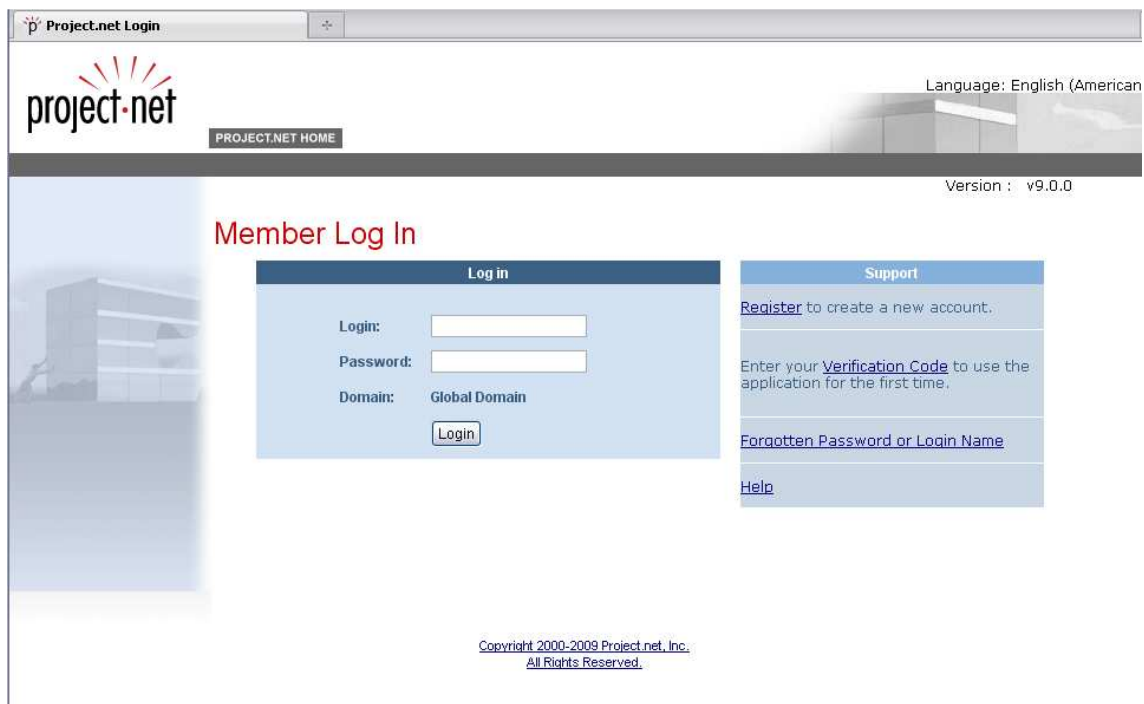


Ilustración 2: Pantalla inicial Project.net