

### 3. Metodología

En este apartado se expone la metodología a seguir para desarrollar el Proyecto, y lograr cada uno de los objetivos planteados. Esta metodología puede dividirse en varias fases:

#### 1. Conocimientos previos.

##### 1.1. Conocimientos previos sobre la tecnología usada por Project.net

Existen una serie de conceptos clave, para llegar a lograr los objetivos planteados. Se ha dicho en el apartado anterior, que Project.net es una aplicación web que corre sobre Apache Tomcat, que éste es un contenedor de servlet, y que la tecnología que usa Project.net es JSP. Vamos a definir a continuación cada uno de estos conceptos [15]:

#### - Servlet

Los Servlets son fragmentos de lógica escritos en Java con una forma definida y que se invocan para generar contenido dinámico y proporcionar un medio para implementar una aplicación en Internet. Todos los Servlet implementan una interfaz denominada Servlet, que define un ciclo vital estándar (una lista de métodos que se invocan de forma predecible). Se inician con el método `init()`, y finalizan con `destroy()`.

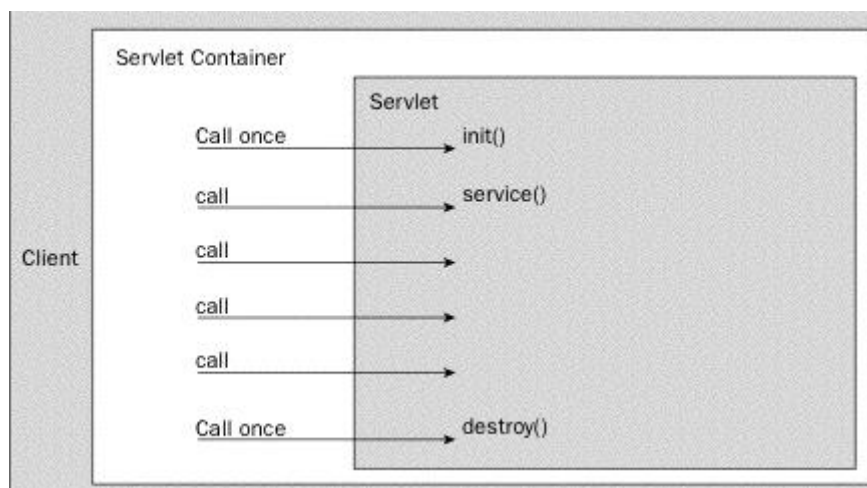


Ilustración 1: Contexto de un Servlet, extraída de [3]

#### - Contenedor de servlets

Entorno de ejecución de servlets.

#### - JSP

Tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. El funcionamiento general de la tecnología

JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del Servlet a generar. Este Servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario. Al igual que los Servlet, las páginas JSP funcionan dentro de un contenedor.

- Aplicación Web

Conjunto de todos los Servlets, páginas JSP y otros archivos relacionados de forma lógica constituye una aplicación Web.

## 1.2. Conocimientos previos sobre Apache Tomcat

Dentro de esta fase también está conocer la arquitectura de Apache Tomcat y cada uno de sus elementos. Los definiremos brevemente a continuación, en el Anexo se profundizará con más detalle en cada uno de ellos.

- El servidor

Es el propio Tomcat.

- El servicio

Un servicio agrupa a un contenedor (normalmente de tipo Motor) con los conectores de dicho contenedor y su componente de nivel superior. Un motor es un componente de procesamiento de solicitudes que representa al motor de Servlet Catalina. Examina los encabezados HTTP para determinar el host o contexto virtual al que deben pasarse las solicitudes.

- Los conectores

Los conectores conectan las aplicaciones a los clientes. Constituyen el punto de recepción de las solicitudes por los clientes y cuentan con un puerto en el servidor. En nuestro caso se utilizará el puerto 7070.

- Las válvulas

Las válvulas son componente que permiten a Tomcat interceptar una solicitud y preprocesarla. Los host, contextos y motores pueden contener válvulas.

- El host

Un host imita la conocida funcionalidad de host virtual de Apache. En Apache, esto permite utilizar varios servidores en un mismo equipo, y distinguirlos por su dirección IP o por su nombre de host.

- El contexto

Por último, nos encontramos a la aplicación Web, también conocida como contexto.

## 2. Implementación Project.net.

La implementación de Project.net se ha hecho en dos máquinas diferentes.

En primer lugar, se hizo la implementación sobre una máquina real, cuyas características principales son las siguientes:

- Microsoft Windows XP Profesional.
- AMD Athlon™ 64x2 Dual.
- Core Processor 5200+, 2.71 Ghz.
- 2.87 GB de RAM.

La segunda de ellas, es una máquina virtual creada con Virtual Box, cuyas características son:

- Microsoft Windows XP Profesional.
- Intel Core i3, 2.2 Ghz
- 512 MB de RAM.
- 10 GB de disco duro (extensible).

Para realizar el proceso de instalación se seguirán las instrucciones descritas en <http://dev.project.net/trac/pnet-community/wiki> que detallaremos más adelante.

Esto ayudará a estudiar la arquitectura de Project.net y a poder hacer un análisis crítico de su configuración, su conexión con otras herramientas, como puede ser una base de datos, o la conexión a un servidor de correo, examinándose en detalle los archivos de configuración que permitirán un mejor entendimiento de todo el funcionamiento de Project.net.

### 3. Evaluación.

Una vez analizada la arquitectura, se pasará a evaluar las características de Project.net. Para ello, se dará de alta en la aplicación a unos usuarios (nombres ficticios), se les asignarán unos roles, unas tareas, se comprobará el correcto uso de las notificaciones mediante correo electrónico. En definitiva, se evaluará el alcance de las funcionalidades de Project.net y su correcto funcionamiento.

Otro de los objetivos planteados, cuyo trabajo también se engloba dentro de esta fase, era la comparación de Project.net con otras aplicaciones afines a nivel documental, Redmine y la solución EPM de Microsoft. Se pensó en la comparación con una de licencia libre y con otra comercial. Redmine fue propuesta por D. Manuel Prado, tutor del Proyecto, y la otra fue elegida por ser una de las más conocidas y populares. La comparación se hará desde dos puntos de vista, el primero de ellos relacionado con los modelos de programación, herramientas asociadas y sus costes, y el segundo, desde el punto de vista de las funcionalidades.

Por último, se evaluará mediante la versión de evaluación del software JProfiler a Apache Tomcat, con el que se calculará el tiempo de respuesta del servidor cada

vez que un usuario intenta acceder a él mediante el protocolo HTTP, y el consumo de CPU de Tomcat en el ordenador en el que esté instalado.