

## CAPÍTULO 2: DISEÑO GLOBAL DEL PROYECTO

En este capítulo explicamos de manera global cómo hemos realizado la implementación del proyecto. Para ello primero vemos por encima las partes que integran el proyecto para tener una idea general. También señalamos las tecnologías que las distintas partes necesitan y terminamos explicando cómo se han realizado dichas partes.

### 2.1.- Visión global del proyecto

El proyecto completo que se ha desarrollado implementa un **servicio Web** completo. En éste contaremos con un cliente que correrá sobre un teléfono móvil, una PDA o cualquier dispositivo de información móvil. El servidor de este **servicio Web** es una aplicación que implementa la especificación SAML. En ella se crearán las entidades que soportan dicha especificación.

Todo el proyecto se divide claramente en dos grandes partes:

- La primera está constituida por el cliente, el cual está desarrollado sobre dispositivo móvil, por lo que trabajará con tecnología específica de este tipo de dispositivos.
- La segunda implementa la aplicación que utiliza la especificación SAML. En ella tendremos que implementar las entidades que proporcionan las ventajas buscadas de la especificación.

### 2.2.- Tecnologías empleadas

El objetivo buscado en este proyecto ha sido el de implementar una solución que trabajase con nuevas tecnologías. De esta forma, en el lado cliente se implementa una nueva solución software que controla al dispositivo y el servidor se hace con una de las soluciones actuales más estables y de mejores propiedades mediante tecnología específica.

Para tener una visión global y más completa del proyecto en conjunto, en lo que se refiere a las tecnologías empleadas, se presenta la siguiente figura en la que se representa de forma esquemática las distintas tecnologías empleadas y el ámbito en el que se mueven.

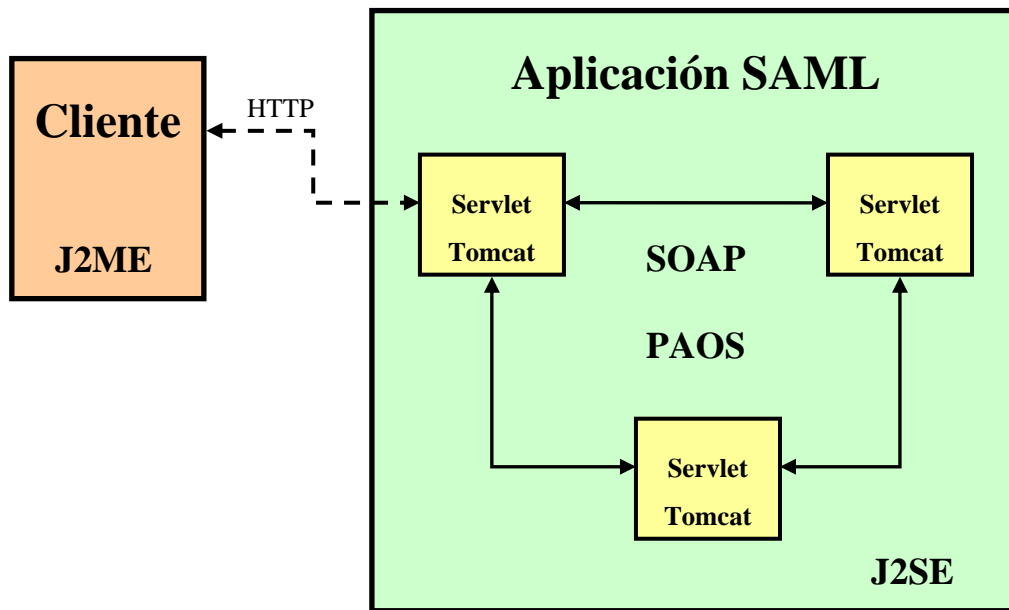


Figura 2. 1: Tecnologías empleadas

En esta figura se pueden ver dos grandes grupos:

- la parte Cliente y
- la aplicación SAML.

Ambos son independientes entre sí, es decir al cliente no le importa como la aplicación esté realizada, solo le interesa que ésta se comunique convenientemente con él; y viceversa. Su estructura interna es la que se ve en la figura. Veamos los distintos aspectos relevantes que hemos mostrado.

- **Cliente:** El cliente está desarrollado sobre un dispositivo móvil, como pueden ser teléfonos móviles, PDAs, etc. La tecnología utilizada es **J2ME**, **Java** para dispositivos móviles.
- **Comunicación Cliente-Aplicación SAML:** Se realiza mediante el protocolo **HTTP**. El cliente le hace una petición de un determinado recurso y la aplicación le devuelve la correspondiente respuesta.
- **Aplicación SAML:** Podemos ver que la aplicación SAML se basa en un par de tecnologías.
  - Por un lado aparecen tres bloques de funcionalidades distintas. Cada uno de ellos realizará una determinada función que la marca **SAML**. Estos bloques se realizan en tecnología **Servlet** que necesita de un servidor **Apache Tomcat**. Se apoyan en tecnología **Java** en concreto **J2SE**.
  - La comunicación entre todos los bloques se realizará mediante los protocolos **SOAP** y **PAOS**. Las entidades crearán los mensajes correspondientes de estos protocolos y lo mandarán mediante **HTTP**.

## 2.3.- Cliente J2ME

El cliente implementado en este proyecto está basado en tecnología **J2ME**, la cual está pensada para ser almacenada sobre dispositivos móviles. El hecho de estar implementado en un dispositivo móvil trae consigo que sus prestaciones sean muy reducidas, por lo que la tecnología **J2ME** no es sencillamente **Java** sobre un dispositivo reducido, sino que tendrá una serie de características específicas debidas al hecho de tener que implementar en un dispositivo tan reducido una tecnología tan potente y amplia como **Java**.

**J2ME** es **Java** para dispositivos móviles. Puesto que estos dispositivos tienen características muy reducidas de potencia de cálculo, de memoria, etc. **J2ME** tendrá que cumplir una serie de restricciones que le harán ser diferente de Java para **J2SE** o **J2EE**.

Sin profundizar demasiado, **J2ME** está basado en tres campos bien diferenciados:

- La **máquina virtual Java** para dispositivos móviles de reducidas prestaciones, KVM, será la encargada de proporcionar una máquina virtual **Java** lo más potente posible en el menor espacio de memoria.
- El **CLDC**, que será el encargado de implementar las funcionalidades **Java** más básicas sobre un dispositivo de reducidas prestaciones.
- El **MIDP**, que implementará funcionalidades de más potencia y de más alto nivel, como pueden ser presentación de gráficos, comunicación de red, almacenamiento de datos en memoria de forma persistente, etc.

Estos tres campos se pueden ver en la figura, donde se puede apreciar claramente el lugar que ocupan en la pila de funcionalidades.

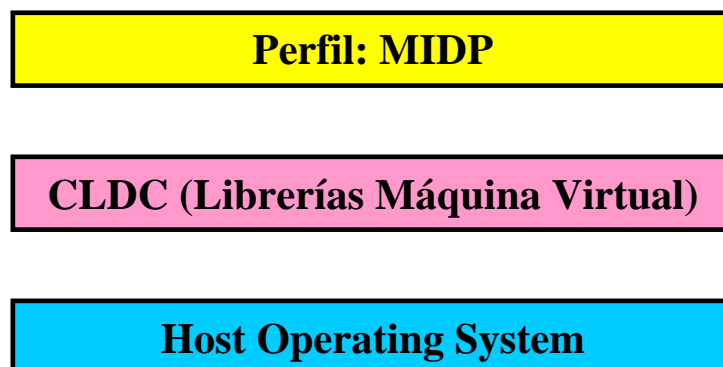


Figura 2. 2: Capas software en J2ME

## 2.4.- Aplicación SAML

A la **aplicación SAML** se conectará el cliente mediante el protocolo **HTTP**. El cliente le realizará una petición de acceso a un recurso existente y ésta le devolverá la correspondiente respuesta.

La **aplicación** consta de tres entidades para proporcionar los servicios de autenticación que proporciona **SAML**. A grandes rasgos podemos decir qué función realiza cada una de las entidades.

- La primera de ellas, que llamaremos **ECP**, es a la que se conecta el cliente. Esta entidad es la encargada de recibir y responder la petición que le llega del cliente.
- Otra de ellas, el **proveedor de identidad (IP)**, es la que proporciona la autenticación de usuario. Valida la identidad del cliente que nos hace la petición. De manera coloquial podemos decir que es la entidad que proporciona la identidad de un determinado cliente.
- La última de ellas, el **proveedor de servicio (SP)**, es la que tiene el recurso que el cliente solicita. Esta entidad necesitará saber si el cliente ha sido autenticado por un proveedor de identidad. Una vez que tiene todos estos datos y conforme a ellos, devuelve una respuesta con un recurso o con un error al ECP. Éste último será el encargado de mandárselo al cliente.

La relación entre ellas las podemos ver en la figura siguiente:

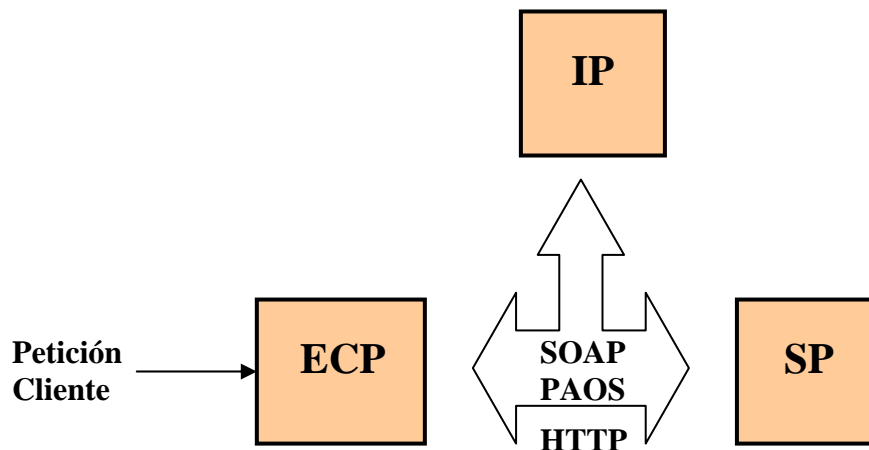


Figura 2. 3: Relación entre entidades de la aplicación SAML

Estas tres entidades las hemos realizado con la misma tecnología. Se trata de la tecnología **Servlet** que permite, mediante hilos independientes, que se pueda dar servicio a los distintos clientes que realicen peticiones. De otra manera, cuando nos llegue una petición para realizar una determinada funcionalidad, se creará una instancia de un hilo que realizará dicha función. Es muy útil para entornos en los que se realicen peticiones a entidades.

Para dar soporte a la tecnología **Servlet** hay que poner un servidor que soporte esta tecnología. Se ha usado el servidor **Apache Tomcat** que proporciona servicio para **Servlets** y para páginas JSP además. Este servidor es el que proporcionará el medio adecuado para que se puedan desplegar los servlet.

La comunicación entre las distintas entidades se realizará mediante los protocolos **SOAP** y **PAOS**. Aunque tengan nombres distintos, son protocolos muy similares, teniendo uno una función que es la inversa del otro. Actualmente, estos protocolos han sido integrados en una **API de Java** por lo que no habrá que introducir sistemas opcionales que proporcionen dichos protocolos. Como vemos en el dibujo los protocolos **SOAP** y **PAOS** van integrados en el protocolo **HTTP** ya que recordamos que la comunicación entre las entidades se realiza en un medio **Web**.